



Transformations de graphes pour les opérations topologiques en modélisation géométrique - Application à l'étude de la dynamique de l'appareil de Golgi

Mathieu Poudret

► To cite this version:

Mathieu Poudret. Transformations de graphes pour les opérations topologiques en modélisation géométrique - Application à l'étude de la dynamique de l'appareil de Golgi. Modélisation et simulation. Université d'Evry-Val d'Essonne, 2009. Français. NNT: . tel-00503818

HAL Id: tel-00503818

<https://theses.hal.science/tel-00503818>

Submitted on 19 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

pour l'obtention du Grade de
DOCTEUR DE L'UNIVERSITÉ D'ÉVRY VAL D'ESSONNE
(Diplôme National - Arrêté du 7 Août 2006)

École Doctorale : **École doctorale Science & Ingénierie**
Secteur de Recherche : **Informatique**

Présentée par :
Mathieu POUDRET

Transformations de graphes pour les opérations topologiques en modélisation géométrique Application à l'étude de la dynamique de l'appareil de Golgi

Thèse soutenue le 15 octobre 2009 devant la Commission d'Examen composée de :

Mohamed Mosbah, Professeur à l'ENSEIRB, LaBRI, Université de Bordeaux	Rapporteur
Pascal Schreck, Professeur à l'Université de Strasbourg, LSIIT	Rapporteur
Agnès Arnould, Maître de conférences à L'Université de Poitiers, XLIM-SIC	Examinateur
Pascal Lienhardt, Professeur à l'Université de Poitiers, XLIM-SIC	Examinateur
Franck Perez, Directeur de Recherche au CNRS, Institut Curie	Examinateur
Olivier Roux, Professeur à l'École Centrale de Nantes, IRCCyN	Examinateur
Pascale Le Gall, Professeur à l'Université d'Évry Val d'Essonne, MAS, École Centrale Paris	Directeur de Thèse

REMERCIEMENTS

"La souris est un animal qui, tué en quantité suffisante et dans des conditions contrôlées, produit une thèse de doctorat." Woody Allen

Peu avant d'entreprendre la rédaction de ce manuscrit, j'ai accroché deux citations sur les murs de mon appartement. La première était une auto-citation, elle disait : "Ta thèse plus efficacement que tout autre tu rédigeras !". J'ai réalisé bien assez tôt que ce crédo serait plus "une sorte de guide, qu'un véritable code". Il m'a cependant aidé à me lever tôt de nombreux - bien trop nombreux ? - matins. La seconde, qui orne encore aujourd'hui les murs de mon appartement, est de ma directrice Pascale Le Gall : "Dans un manuscrit de thèse comme dans le scénario d'un film, chaque petite chose doit avoir une raison d'être là. Rien n'est laissé au hasard." Ce conseil compte parmi les nombreux enseignements de Pascale. Ainsi, pour ces nombreuses petites phrases pleines de sens, pour cet art qu'elle a de prendre souvent les bonnes décisions un coup avant tout le monde et enfin pour s'être trompée de catégorie au téléphone ce jour là, je tiens à remercier chaleureusement Pascale. Je tiens également à remercier Agnès Arnould pour la générosité de son encadrement tout au long de ma thèse. En particulier, pour m'avoir aidé à me jeter à l'eau lorsqu'il s'agissait de rédiger, puis de re-rédiger les nombreux objets mathématiques qui peuplent ce manuscrit. Enfin, je remercie Agnès également pour sa bienveillance et son écoute les nombreuses fois où nos discussions dépassaient le strict cadre de nos travaux.

La rédaction de ce mémoire de thèse de doctorat n'aurait pas été possible sans le soutien des membres du groupe de travail GolgiTop, qui se réunit régulièrement dans le cadre des ateliers du Programme d'Épigénomique (Génopole, Évry). Je tiens en particulier à remercier Jean-Paul Comet (laboratoire I3S, Université de Nice Sophia Antipolis) pour nos réunions de travail mémorables en haut de la tour Évry 2, où armés de quelques marqueurs et d'une brosse usée, nous réinventions nos publications avec passion. Je remercie également Philippe Meseure (laboratoire XLIM-SIC, Université de Poitiers), qui a su veiller à ce que je garde un pied sur terre lorsque je m'envolais dans les méandres parfois obscures des méthodes formelles. Enfin, la rédaction de la seconde partie de ce manuscrit aurait été bien difficile sans le support des biologistes du groupe GolgiTop : Jean-Marc Verbavatz, Alain Rambourg (CEA, Saclay) et François Képès (Programme d'Épigénomique), qui ont su me transmettre leur savoir avec une aisance incomparable.

Je remercie également Michel Bauderon, Olivier Ly et Mohamed Mosbah (laboratoire LaBRI, Université de Bordeaux), que j'ai eu la chance de rencontrer lors d'une réunion de travail à Bordeaux. Ils m'ont fourni une aide précieuse au sujet de l'utilisation des transformations de graphes, essentielles dans ce manuscrit. Enfin, je tiens à remercier Berthold Hoffmann (département Mathématiques et Informatique de l'Université de Bremen, Allemagne) pour l'aide précieuse qu'il m'a apporté sur l'utilisation des variables dans ces mêmes transformations.

Il va de soi que dans les moments les plus difficiles du doctorat, l'aide de la famille, des collègues (et néanmoins amis) et amis (qui eux pour le coup ne sont pas des collègues) est indispensable. P'pa, M'man, ma Bidounette, Angèle, Nico, Alicette et mon petit Lubinciraptor, merci beaucoup pour ces nombreuses fois où je n'étais pas là, où encore pour ces nombreuses autres fois où je n'étais qu'à moitié

là. Et surtout, merci pour m'avoir souvent soutenu avec un gâteau de petits beurrés, ou bien un bon film, lorsque j'étais las. Merci également aux cousins et cousines avec qui j'ai passé de mémorables soirées.

Merci à Seb H. pour les soirées FPS ! Merci à Martin D. pour tes grands moments de solitude... Navré, mais Patrick Bouchitey n'est toujours pas à l'affiche d'Armageddon. Merci à Pef L. pour m'avoir appris à me taire de peur d'en prendre une. Merci aux pas-presque-déjà-femmes des trois lascars précédemment cités. Merci à Olivier "le Hunter" T. pour ses nombreux conseils de drague. Merci à Roro P. et Bobo A. pour avoir eu le courage de faire une thèse avec Daniel M. (Daniel, les réponses à tes deux questions sont OUI et OUI !). Merci à Carlos P. Philippe II d'Espagne a annexé le Portugal en 1580 ; à quelques siècles près j'avais raison. Merci à Samuel "Barney" P. pour les inoubliables séances de TP en GPhy. Tu remercieras également ta femme, sans qui ma vie d'aujourd'hui ne serait pas tout à fait la même. Merci à Olivier "mon Fillot" K. pour sa bonne humeur permanente. Merci à Sylvie D. et Françoise P. pour avoir prouvé que l'humain diffuse l'information plus rapidement que la machine. Merci à Éric A. ; Éric, je me dépêche de terminer mes remerciements avant que mon Mac ne prenne feu. Merci enfin à Sheepbox et à leurs groupies ! J'espère qu'avec ça j'aurais accès à vie au backstage de vos concerts !

Merci à mon Ti'Fred H. et à "qu'au" Élodie A. pour avoir toujours une vache qui rit (marque déposée par Fromageries Bel) pour moi dans leur frigo. Merci à Sebastien P., Erika S. et Liam (mais au fait, quel est son nom ?), même s'ils sont parfois affreusement loin. Merci à Ben "Eskimo" L. ; tu vois, finalement je n'ai pas redoublé ! Merci à Aurélie "Kiki" K. pour ses nombreux Kikismes. Kiki, un Van Gogh coûte généralement plus de 1500 euros et non, Jean-Jacques Rousseau n'est pas client chez Leclerc. Merci à Angel "La vieille" P. pour ses incroyables potins. Merci également à Laurane G. pour nos mémorables soirées confidences. Merci à la bande à Rhum1 pour les soirées arrosées à la brioche Pasquier. Merci à l'ensemble des théâtraux de Chantiers Théâtre. En particulier, merci à Jean-Pierre G. pour avoir cru en moi. Merci à Anaïs "Clic-Clac" V. pour New-York, ta grosse valise, tes gâteaux et nos belles soirées tous les deux. Enfin, merci à Nicolas "Tchoutchou" B. pour avoir dit un jour que "nous serons toujours un peu jaloux l'un de l'autre".

Pour ceux qui ont scruté avidement ces remerciements en espérant y voir leur nom, mais que j'aurais oublié, inscrivez votre nom dans le blanc prévu à cet effet. Je remercie - - - -, sans qui l'ensemble de ces travaux n'auraient pu voir le jour.

Encore merci à tous.

SOMMAIRE

I Transformations de graphes pour les opérations topologiques en modélisation géométrique XVII

1	Modélisation géométrique à base topologique	1
1.1	Modélisation géométrique	2
1.2	Modélisation géométrique à base topologique	4
1.3	Cartes généralisées	7
1.3.1	Un premier exemple	8
1.3.2	Pré-G-cartes	10
1.3.3	Contraintes de cycles	13
1.4	Opérations de base	15
1.4.1	Ajout d'un brin	16
1.4.2	Suppression d'un brin	16
1.4.3	Couture	17
1.4.4	Décousure	20
1.5	Opérations de haut niveau	21
1.5.1	Cône	21
1.5.2	Chanfreinage	23
2	Transformations de graphes et cartes généralisées	27
2.1	Transformations de graphes	28
2.1.1	Principe général	30
2.1.2	Principales approches	30
2.2	Approche par double somme amalgamée	32
2.2.1	Transformation directe	32
2.2.2	Condition d'arcs pendants	35
2.3	Transformations de cartes généralisées	37
2.3.1	Règles d'ajout et de suppression d'un brin isolé dans une G-carte	38
2.3.2	Satisfaction de la contrainte d'arcs incidents	39
2.3.3	Préservation des pré-G-cartes	43
2.3.4	Satisfaction de la contrainte de cycles	44
2.4	Conclusion	47
3	Transformations de graphes avec variables et cartes généralisées	49
3.1	Principe général de la transformation de graphes avec variables	50
3.1.1	Règles de transformation avec variables	50
3.1.2	Instanciation d'une règle de transformation avec variables	51
3.1.3	Application d'une règle de transformation avec variables	51
3.1.4	Variables graphes et clones	53

3.2	Variables orbites	57
3.2.1	Règles de transformation avec variable orbite	57
3.2.2	Instanciation des variables orbites	59
3.3	Renommages d'orbites	68
3.3.1	Motivations	68
3.3.2	Renommages d'arcs dans les graphes	72
3.3.3	Renommages et effacements d'arcs dans les graphes	73
3.3.4	Règles de transformation avec renommages d'orbites	75
3.3.5	Instanciation des renommages d'orbites	77
3.4	Renommage d'orbites et cohérence des cartes généralisées	88
3.4.1	Contrainte d'arcs incidents	88
3.4.2	Préservation des pré-G-cartes	92
3.4.3	Condition d'arcs pendants	94
3.4.4	Contrainte de cycles	94
4	Perspectives : transformation de graphes et cartes généralisées plongées	105
4.1	Modélisation des jonctions communicantes	106
4.2	Cartes généralisées plongées	107
4.3	Variables attributs et plongement	108
4.3.1	Règles de modification de plongement	108
4.3.2	Préconditions de plongement	110
4.3.3	Contrainte de plongement	111
4.4	De l'utilité de manipuler plusieurs variables orbites	112
II	Application à l'étude de la dynamique de l'appareil de Golgi	117
5	L'appareil de Golgi	119
5.1	La voie sécrétoire	120
5.2	Trois modèles pour l'appareil de Golgi	123
5.2.1	Modèle vésiculaire	123
5.2.2	Modèle de la maturation	126
5.2.3	Modèle du flux continu de membranes	128
5.3	Bilan	131
6	Méthode de discrimination pour l'appareil de Golgi	133
6.1	Formulaire	134
6.2	Notre méthode de discrimination de modèles	135
6.2.1	Discriminer par simulation	135
6.2.2	Méthodologie de discrimination de modèles	137
6.3	Modèle vésiculaire	138
6.3.1	Le modèle initial	138
6.3.2	Mise à jour du modèle vésiculaire	144
6.3.3	De la nécessité d'une zone tubulaire	148
6.3.4	Bilan	153
6.4	Modèle de la maturation	154
6.4.1	Un transport rétrograde coûteux	154
6.4.2	Bilan	156
6.5	Modèle du flux continu de membranes	156
6.5.1	De l'insuffisance de l'étude des rapports de volume sur surface	156
6.5.2	Représentations tridimensionnelles des modèles de l'appareil de Golgi	159
6.5.3	Validation des modèles tridimensionnels par des coupes	160

6.5.4	Bilan	162
6.6	Conclusion de notre étude	162
7	Perspectives : vers un simulateur biologique à base topologique	165
7.1	La simulation des compartimentations biologiques	166
7.1.1	Simulation et modèle	166
7.1.2	Quelques approches pour la simulation des compartimentations biologiques . . .	167
7.2	Vers un simulateur biologique à base topologique	172
7.2.1	Deux modèles biologiques	172
7.2.2	Le noyau de 3-G-cartes	174
7.2.3	Plongements et sorties graphiques	180
7.2.4	L'implantation des règles de transformation	185
7.2.5	Bilan et perspectives	197
A	Rappels et notations	207
A.1	Les mots	207
A.2	Théorie des graphes	208
A.2.1	Graphes	208
A.2.2	Morphismes de graphes	209
A.2.3	Notions complémentaires	211
A.3	Théorie des catégories	212
A.3.1	Catégories	213
A.3.2	Diagrammes commutatifs	213
A.3.3	Sommes amalgamées	213
A.3.4	Produits fibrés	215
B	Plongements du simulateur	219
B.1	Plongement des demi-faces	219
B.2	Plongement des arêtes simples	220
B.3	Plongement des 0-cellules	222
B.4	Exemple du bourgeonnement des vésicules	222
	Bibliographie	225

INTRODUCTION

Le principal objectif des travaux développés dans ce manuscrit est de proposer un langage à base de règles permettant d’animer des structures topologiques. Afin d’illustrer notre langage, nous appliquons notre système d’animation à des objets biologiques cellulaires où la structuration en compartiments apparaît comme étant essentielle. Ainsi, l’étude menée dans ce manuscrit se situe à la croisée de trois domaines de l’informatique. Les structures de données utilisées en *modélisation géométrique à base topologique* nous permettent de représenter la structure des objets que nous souhaitons animer. Nous nous appuyons également sur les *transformations de graphes*, qui sont des outils issus des méthodes formelles, afin de fournir un cadre pour l’écriture de notre langage d’animation. Enfin, puisque nous appliquons notre système d’animation à des objets biologiques structurés, nous nous insérons dans le cadre de la *modélisation des systèmes biologiques compartimentés*.

Contexte

La modélisation géométrique à base topologique

De la thèse de Sutherland en 1963 [Sutherland 63] – dans laquelle une méthode permettait de dessiner sur un écran – aux plus récents succès cinématographiques – dans lesquels images virtuelles et prises de vue réelles se mélangent à l’insu du spectateur – la *modélisation géométrique* a connu une croissance sans équivoque. Elle couvre aujourd’hui de nombreux domaines d’application tels que la conception assistée par ordinateur, les jeux vidéos, l’architecture, ou même les sciences du vivant. Ce domaine de l’informatique graphique consiste à proposer des structures de données afin de représenter graphiquement des objets – souvent issus du monde réel – dans un ordinateur. Dans ce manuscrit, nous nous intéressons à une branche particulière de la modélisation géométrique : la *modélisation géométrique à base topologique* [Bechmann 07b].

En modélisation géométrique à base topologique, on fait le choix de distinguer la structure et la forme des objets. À ce titre, dans la littérature, on parle respectivement de la *topologie* – ou encore *structure topologique* – et de la *géométrie* des objets. Un exemple est proposé en Fig. 1. Une manière de décrire la structure d’une table consiste à dire qu’elle est composée de quatre pieds P_1 à P_4 connectés à un volume V . Cette topologie est modélisée par le graphe de gauche sur la figure. Remarquons que dans une approche purement topologique, la forme de même que la position des différentes parties de la table importe peu ; ces deux données sont d’ordre géométrique. Deux géométries possibles de la table – on parle également de plongement géométrique – sont illustrées en Fig. 1. Dans le cadre de ce manuscrit, nous manipulons principalement la topologie des objets. La géométrie sera mentionnée dans ce manuscrit, notamment au moment de l’application à la biologie, sans qu’elle soit au cœur de notre étude.

Remarquons que de nombreux *modèles topologiques* ont été définis et permettent de représenter la topologie des objets. Parmi ceux-ci, nous choisissons les *cartes généralisées* [Lienhardt 89, Lienhardt 94]. Ce modèle présente l’avantage d’être défini de manière homogène en toutes dimensions. De plus, de

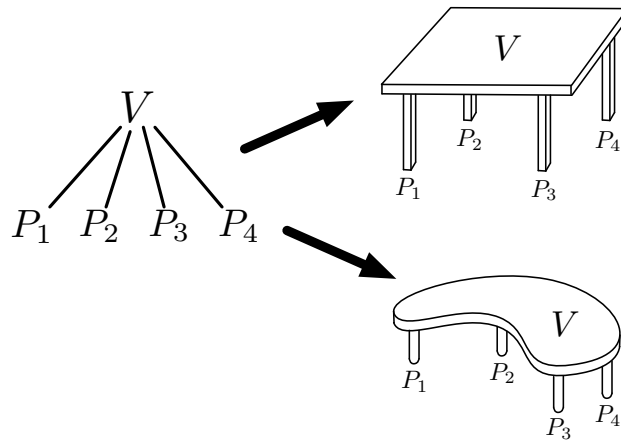


FIGURE 1 – Distinction entre la topologie et la géométrie

nombreuses opérations topologiques ont été définies sur les cartes généralisées. Elle permettent de construire, et donc de modifier, des structures topologiques.

Les méthodes formelles pour la modélisation géométrique

Les méthodes formelles sont des outils du génie logiciel qui présentent l'avantage d'introduire une grande rigueur dans le cycle de développement d'un logiciel. Notamment, elles permettent d'accroître leur robustesse, de réduire le temps de développement et de faciliter les activités de maintenance. De plus, bien qu'il n'en soit pas fait mention dans ce manuscrit, les méthodes formelles favorisent l'utilisation en amont d'outils de vérification et de test. Les principaux logiciels développés en modélisation géométrique sont les modeleurs [Ble , 3ds , May]. À partir d'une structure de données permettant de représenter les objets, les modeleurs fournissent un ensemble d'opérations qui permettent de manipuler les objets, on parle souvent d'opérations de construction ou d'édition. De nos jours, les modeleurs sont des logiciels imposants qui nécessitent de nombreuses hommes-années de développement. Dans le cadre de nos travaux nous souhaitons profiter des méthodes formelles afin d'aider à automatiser leurs développements.

Deux principales caractéristiques des structures de données de la modélisation géométrique à base topologique – on parle également de *modèles topologiques* – ont conduit la communauté d'informatique graphique à s'intéresser aux méthodes formelles. En premier lieu, les modèles topologiques sont généralement des structures simples et régulières proches des modèles mathématiques utilisés par les méthodes formelles. De plus, les modèles topologiques sont des structures de données munies d'invariants qui assurent la cohérence topologique des objets. De fait, ces structures favorisent l'emploi des spécifications algébriques. Remarquons que dès 1992, la spécification algébrique [Bertrand 92] du modeleur Topofil [Bertrand 98] a permis à Y. Bertrand de réduire considérablement la taille du code. Remarquons toutefois que Y. Bertrand n'a pas utilisé d'outil spécifique des méthodes formelles lors de ses travaux. Intuitivement, la simple écriture de la spécification de son modeleur lui a permis d'implanter Topofil de manière relativement rapide.

Par la suite, l'utilisation de l'outillage classique des méthodes formelles – à savoir le prototypage, la génération de tests, le raffinement et l'étude de propriétés – a été mis en œuvre dans le cadre d'autres travaux centrés sur la modélisation géométrique. Cependant, que ce soit dans le cadre des spécifications formelles en CASL [Cas , Ledoux 01] ou de la méthode B [Ledoux 02b], l'outillage s'est révélé difficile à manipuler. En particulier, les outils de prototypage et de génération de tests ne semblent pas assez

matures. En effet, ceux-ci ne permettent pas encore de travailler sur de très grandes données, telles que celles utilisées en modélisation géométrique à base topologique. Remarquons que des travaux parallèles s'intéressent également à l'utilisation de logiciels de preuves formelles tels que Coq [Coq] dans le but de formaliser des théorèmes de la modélisation géométrique [Magaud 08]. Dans le cadre de ce manuscrit, notre principale motivation est la conception de modeleurs assistée par les méthodes formelles dans le but d'automatiser les cycles de conception et de développement.

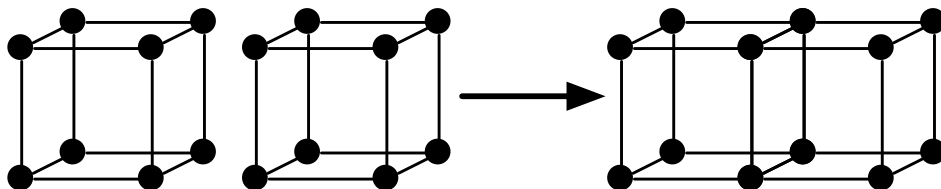


FIGURE 2 – Une règle de transformation de graphes

Nous choisissons un outil plus opérationnel, et notamment plus visuel : les *transformations de graphes* [Ehrig 06]. Les transformations de graphes sont inspirées des systèmes de réécriture de termes utilisés pour le prototypage de spécifications algébriques. Le domaine des transformations de graphes propose un langage formel basé sur des règles de transformation qui permettent de modifier des graphes. Ces règles présentent l'avantage de décrire de manière abstraite – et indépendante de l'implantation – les modifications précises que l'on souhaite apporter à un graphe. Un exemple de règle est donné en Fig. 2, celle-ci permet de coller deux graphes de forme cubique le long d'une face. Dans ce manuscrit, nous proposons une première étude de l'apport des transformations de graphes en modélisation géométrique à base topologique.

La modélisation de structures biologiques compartimentées

Dans de nombreux domaines d'application, la structure des objets joue un rôle essentiel. C'est le cas en biologie, où de nombreux mécanismes sont le fait de transferts de molécules entre des *compartiments*. Au sein d'une compartimentation, la topologie nous renseigne sur l'agencement des compartiments entre eux. Remarquons que bien souvent, en biologie, la structure elle-même est dynamique. Nous pouvons prendre pour exemple l'un des phénomènes fondateurs de la biologie cellulaire [Alberts 02] : la mitose (voir Fig. 3). Ici, une cellule biologique – que l'on peut abstraire par un compartiment – est divisée en plusieurs cellules filles. Notons que ce mécanisme bien connu est à la base de phénomènes biologiques plus complexes reposant notamment sur des variations de la topologie. Ainsi, lors de la morphogenèse, les divisions successives de cellules initialement identiques conduisent au développement de compartiments spécialisés qui présentent des topologies variées : les organes.

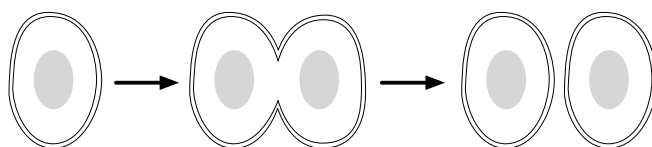


FIGURE 3 – La mitose cellulaire

De nombreux outils s'intéressent à l'animation de modèles biologiques compartimentés à l'aide de règles de transformation. En effet, la modélisation de systèmes dynamiques par des règles présente plusieurs avantages. Tout d'abord, les règles permettent de décomposer l'évolution du système en isolant

par une règle chacun des mécanismes qui prennent part à cette évolution. De plus, en biologie, les modélisations à base de règles sont particulièrement adaptées puisque, notamment, les biologistes décrivent l'ensemble des réactions chimiques par des règles. Remarquons que jusqu'à présent, les modélisations à base de règles dédiées à la biologie sont déconnectées de la modélisation géométrique à base topologique. En particulier, les compartiments sont représentés de manière purement abstraite [Calzone 06, Regev 04, Cardelli 05]. Ce mode de représentation permet de modéliser des molécules qui sont transportées d'un compartiment à un autre, mais la prise en compte de la dynamique des compartiments est souvent limitée : on s'intéresse principalement aux mécanismes de division, de fusion et d'inclusion de compartiments.

Objectifs

Transformations de graphes pour les opérations topologiques en modélisation géométrique

Dans ce manuscrit, nous voyons les cartes généralisées comme des graphes particuliers sur lesquels des *contraintes de cohérence* assurent la cohérence topologique des objets. Ainsi, nous nous plaçons dans le contexte des graphes et proposons de traduire les opérations des cartes généralisées en règles de transformation de graphes. Nous spécialisons alors les règles de transformation de graphes pour la modélisation géométrique à base topologique. Pour cela, nous étendons la syntaxe des règles afin de proposer une manière intuitive d'écrire les opérations, puis définissons des *critères syntaxiques* qui assurent la cohérence des objets produits par l'application des règles. Ainsi, les objets résultats de l'application des règles sont bien des cartes généralisées.

La traduction des opérations en règles de transformation constitue la pierre angulaire de ce manuscrit. Dans l'approche classique, l'introduction d'une nouvelle opération suit deux étapes essentielles. Dans un premier temps, il s'agit de proposer une définition mathématique de l'opération. Dans un second temps, on doit prouver que les objets produits par l'opération satisfont les contraintes de cohérence des cartes généralisées. Nous proposons un langage qui permet d'écrire les opérations directement sous forme de règles. Une grande partie de ce manuscrit est ainsi consacrée à l'introduction de la syntaxe de nos règles. Cette syntaxe doit être suffisamment expressive pour couvrir l'ensemble des modifications de structures mises en jeu dans les opérations. Par rapport à l'approche classique, notre langage présente un avantage majeur. En effet, il nous permet d'assurer la satisfaction des contraintes de cohérence par construction, à l'aide de critères syntaxiques qui restreignent l'écriture des règles.

Application à l'étude de la dynamique de l'appareil de Golgi

L'application de nos travaux à la biologie a pour origine l'étude d'un mécanisme biologique dans lequel le rôle joué par la structure apparaît également essentiel. Une représentation schématique d'une cellule biologique est proposée en Fig. 4. La voie sécrétoire de la cellule est le chemin suivi par les protéines depuis la périphérie du noyau (voir l'annotation 2 sur la figure) – où elles sont synthétisées – jusqu'au milieu extra-cellulaire. Les biologistes qui étudient la voie sécrétoire s'accordent à dire que parmi les acteurs de l'excrétion des protéines, l'*appareil de Golgi* [Képès 06, Farquhar 81, Mironov 97, Rambourg 79, Mollenhauer 91] joue un rôle primordial. Cependant, en raison de limites techniques liées à l'observation, la structure spatiale de l'appareil de Golgi reste méconnue. Ainsi, sur la plupart des images issues de la microscopie électronique, l'appareil de Golgi apparaît comme une pile de citernes lisses et déconnectées (voir l'annotation 6 sur la figure). En revanche, d'autres observations suggèrent que l'appareil de Golgi est constitué de régions tubulaires qui connectent les saccules. Ainsi, les différentes observations sont à l'origine de plusieurs modèles de l'appareil de Golgi qui se distinguent principalement par leur structure, fondamentalement différentes. L'un des objectifs de cette thèse est de poser les bases d'un outil d'aide à la discrimination de ces différents modèles de l'appareil de Golgi. Intuitivement, il s'agit d'aider les biologistes de l'appareil de Golgi à déterminer quel modèle est le plus

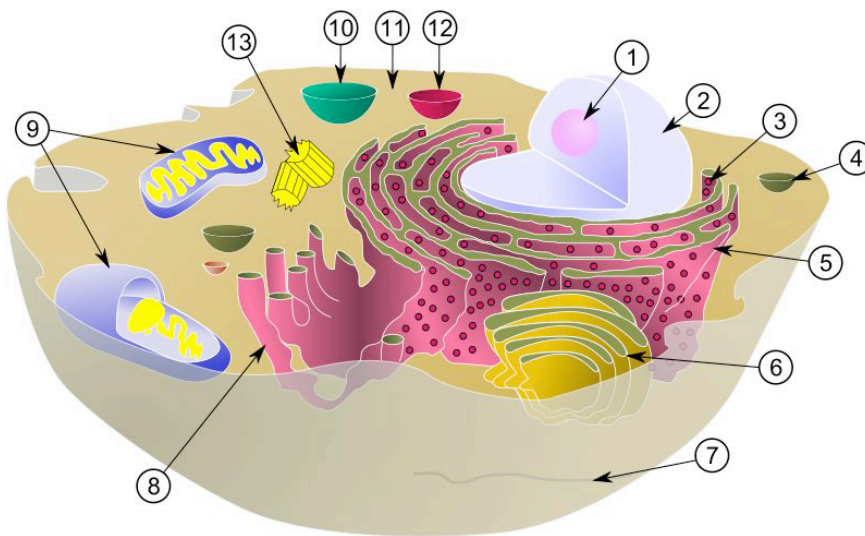


FIGURE 4 – Une cellule biologique. Le nucléole (1), le noyau (2), un ribosome (3), une vésicule (4), le réticulum endoplasmique rugueux (ou granuleux) (5), l'appareil de Golgi (6), un microtubule (7), le réticulum endoplasmique lisse (8), deux mitochondries (9), la vacuole (10), le cytoplasme (11), un lysosome (12), le centrosome (13). Source : [Wik]

proche des observations.

Dans ce manuscrit, nous proposons une méthodologie de discrimination basée sur la simulation des différents modèles de l'appareil de Golgi. Notamment, nous proposons un premier prototype de simulateur dans lequel nous utilisons notre langage de transformation de structures topologiques afin d'animer les modèles biologiques. Remarquons que notre langage à base de règles de transformation nous place dans le contexte des modélisations à base de règles dédiées à la biologie. Cependant, nous nous démarquons des travaux évoqués précédemment en ce sens que nous proposons de représenter explicitement les compartiments biologiques à l'aide de cartes généralisées. Une telle représentation nous renseigne sur le voisinage des compartiments. Par exemple, pour un compartiment donné, nous savons notamment dire s'il est connecté ou non à d'autres compartiments. Notre approche présente l'avantage d'allier la puissance de la modélisation géométrique à base topologique, à l'expressivité et à la facilité de manipulation des langages à base de règles. Remarquons que ce besoin d'animer des structures topologiques de la modélisation géométrique à l'aide de règles, a déjà émergé par ailleurs dans d'autres domaines d'application tels que la géologie [Léon 08].

Organisation du manuscrit

La première partie de ce manuscrit est consacrée à la formalisation des opérations des cartes généralisées. Elle est divisée en quatre chapitres. Dans le chapitre 1, nous présentons la modélisation géométrique à base topologique. Nous donnons un bref aperçu des différentes classes de modèles topologiques et introduisons les cartes généralisées comme étant des cas particuliers de graphes. À ce titre, nous introduisons les contraintes de cohérence des cartes généralisées comme étant des contraintes sur des graphes. Dans ce chapitre, nous présentons également les quatre opérations de base des cartes généralisées. Nous introduisons également deux opérations plus évoluées : nous proposons une définition de l'opération de cône et présentons intuitivement l'opération de chanfreinage.

Dans le chapitre 2, nous présentons dans un premier temps différentes approches de la transformation de graphes. Nous introduisons ensuite en détail l’approche par *double somme amalgamée*, sur laquelle nous nous appuyons dans ce manuscrit. Nous montrons de quelle manière les règles de transformation de graphes nous permettent d’appliquer des opérations aux cartes généralisées. À ce titre, nous introduisons un critère syntaxique sur les règles afin de nous assurer que lorsque nous transformons une carte généralisée à l’aide d’une règle, les contraintes de cohérence des cartes généralisées restent satisfaites. En nous appuyant sur l’opération de couture des cartes généralisées, nous montrons que l’approche classique de la transformation de graphes par double somme amalgamée ne permet pas de formaliser – en l’état – les opérations évoluées telles que l’opération de cône.

La traduction des opérations des cartes généralisées en règles de transformation de graphes nécessite d’introduire des variables dans nos règles. Ainsi, dans le chapitre 3, nous commençons par dresser un bref état de l’art des différentes variables introduites par B. Hoffmann [Hoffmann 05]. En nous inspirant des variables clones et des variables graphes, nous proposons des variables orbites spécialisées pour les règles de transformation de cartes généralisées. Ces variables nous permettent de dupliquer des orbites – c’est-à-dire des sous-graphes – dans une carte généralisée. Dans un premier temps, nous montrons que des règles de transformation munies de telles variables nous permettent de traduire les opérations de couture et de décousure. Dans un second temps, nous étendons ces règles à l’aide d’opérations de renommage et de suppression d’arcs dans des orbites. Cette extension nous permet d’augmenter le pouvoir d’expression de nos règles, afin de traiter des opérations topologiques de plus haut niveau telles que l’opération de cône et l’opération de chanfreinage. Ceci nous mène à étendre notre critère syntaxique aux règles de transformation munies d’opérations de renommage et de suppression d’arcs dans des orbites.

Dans le chapitre 4, qui clôt la première partie de ce manuscrit, nous donnons quelques pistes afin de permettre l’introduction de données qui ne sont pas d’ordre topologique, dans nos règles de transformation. En modélisation géométrique, on parle de *plongement* pour désigner ces données. Ce chapitre s’articule autour d’un exemple biologique, les jonctions communicantes [Alberts 02], qui nous permet d’introduire deux types de plongement : un *plongement géométrique* et un *plongement biochimique*. Le plongement géométrique nous permet de donner une forme et une position à nos objets tandis que le plongement biochimique permet d’associer des protéines aux compartiments. Après avoir introduit intuitivement la notion de cartes généralisées plongées, nous montrons comment les variables attributs définies par B. Hoffman nous permettent d’écrire des règles qui manipulent les plongements. Ces variables – nous pouvons dès lors parler de variables de plongement – nous mènent à étendre le critère syntaxique qui assure la cohérence des objets transformés par nos règles.

Dans la seconde partie de ce manuscrit, nous nous focalisons sur l’étude de la dynamique de l’appareil de Golgi. Cette partie est divisée en trois chapitres. Le chapitre 5 est un chapitre de biologie. Il introduit l’appareil de Golgi ainsi que les différentes observations qui ont conduit les biologistes à proposer différents modèles de l’appareil de Golgi. Nous y introduisons les trois modèles étudiés dans la seconde partie du manuscrit : le modèle vésiculaire, le modèle de la maturation et le modèle du flux continu de membranes.

Dans le chapitre 6, nous présentons notre méthodologie de discrimination des trois modèles de l’appareil de Golgi. Nous nous focalisons ensuite sur l’étude des rapports de volume sur surface de leurs compartiments et montrons ainsi que le modèle vésiculaire ne correspond pas aux observations faites par les biologistes. Par une étude analogue, nous montrons une limite du modèle de la maturation. Enfin, à partir de représentations tridimensionnelles des différents modèles, nous montrons que le modèle du flux continu de membranes est celui qui semble s’approcher le plus des connaissances biologiques.

Dans le chapitre 7, nous établissons le cahier des charges d’un simulateur axé sur l’animation de modèles biologiques, à l’aide des règles de transformation proposées dans la première partie du manus-

crit. Ce chapitre s'appuie sur le développement d'un premier prototype de simulateur. Ainsi, après avoir dressé un bref état de l'art de la modélisation informatique en biologie, nous introduisons les différentes structures de données ainsi que les principaux algorithmes à la base de notre prototype. Pour cela, nous nous appuyons sur deux modèles biologiques inspirés de l'appareil de Golgi.

Guide de lecture

La première partie de ce manuscrit s'appuie sur quelques notions classiques issues de l'informatique théorique. Ainsi, les définitions classiques de même que certaines notations utilisées dans ce manuscrit sont données en annexe A. L'annexe A est organisée de la manière suivante :

- en annexe A.1, nous proposons quelques rappels sur les *mots* et leur notations. Dans le manuscrit, nous utilisons des mots construits sur l'alphabet des étiquettes qui dénotent les différents liens de voisinage qui existent au sein d'une structure topologique ;
- en annexe A.2, nous proposons quelques rappels au sujet de la théorie des *graphes*. Nous introduisons les notions de *graphe partiellement étiqueté* et de *graphe non orienté* – qui nous permettent de définir les cartes généralisées comme des cas particuliers de graphes – et les notions de *morphisme de graphes partiellement étiquetés* et de *morphisme d'inclusion de graphes partiellement étiquetés*, que nous utilisons au sein des règles qui transforment les graphes. Nous rappelons également les notions de *chemin*, de *cycle*, de *sous-graphe généré* et de *soustraction de deux graphes partiellement étiquetés* ;
- enfin, en annexe A.3, nous rappelons quelques notions de la théorie des *catégories* telles que les *diagrammes commutatifs*, les *sommes amalgamées* et les *produits fibrés*. Ces différentes notions nous permettent notamment de coller des graphes entre eux et d'effectuer des produits de graphes.

Nous invitons ainsi le lecteur à se reporter aux annexes lorsque nous utilisons ces différentes notions dans le corps du manuscrit.

Première partie

Transformations de graphes pour les opérations topologiques en modélisation géométrique

MODÉLISATION GÉOMÉTRIQUE À BASE TOPOLOGIQUE

Sommaire

1.1	Modélisation géométrique	2
1.2	Modélisation géométrique à base topologique	4
1.3	Cartes généralisées	7
1.3.1	Un premier exemple	8
1.3.2	Pré-G-cartes	10
1.3.3	Contraintes de cycles	13
1.4	Opérations de base	15
1.4.1	Ajout d'un brin	16
1.4.2	Suppression d'un brin	16
1.4.3	Couture	17
1.4.4	Décousure	20
1.5	Opérations de haut niveau	21
1.5.1	Cône	21
1.5.2	Chanfreinage	23

Dans ce chapitre, nous donnons un bref aperçu de la *modélisation géométrique* puis présentons ce qui caractérise la *modélisation géométrique à base topologique*. Nous introduisons ensuite le modèle des *cartes généralisées*, sur lequel nous nous appuyons tout au long de ce manuscrit. Afin de nous placer dans le cadre de la transformation de graphes, nous définissons les cartes généralisées comme étant des graphes spécialisés par des contraintes d'ordre topologique. Enfin, nous présentons quelques-unes des opérations définies sur les cartes généralisées. Nous introduisons dans un premier temps les opérations de base qui permettent de modifier localement une carte généralisée, puis donnons deux exemples d'opération de haut niveau – le *cône* et le *chanfreinage* – qui permettent d'effectuer des modifications plus globales.

1.1 Modélisation géométrique

La *modélisation géométrique* est un domaine bien connu de l'informatique graphique [Sutherland 63] où l'on s'intéresse à la construction d'objets virtuels à l'aide de logiciels spécialisés appelés *modeleurs*. Ces logiciels permettent tant la construction des objets que leur manipulation (modification de leur structure, déformation, etc.). La modélisation géométrique est utilisée dans de nombreux domaines d'application. En architecture, on utilise la modélisation géométrique pour construire des modèles de bâtiments. Ces modèles permettent de simuler la manière dont les ondes – notamment les ondes lumineuses [Horna 09] – se propagent à l'intérieur d'un bâtiment. En géologie, la modélisation géométrique est utilisée afin de décrire l'évolution d'un sous-sol [Léon 08]. Remarquons que les techniques de modélisation géométrique précèdent généralement une étape de rendu [Bechmann 07a], qui permet d'effectuer un affichage des objets. Pour cela, on plaque des textures – c'est-à-dire des images (par exemple du bois ou bien du granite) – sur les objets. Il est également possible de définir finement, pour chaque point de l'objet, de quelle manière la lumière est ré-émise (l'objet est-il mat ? la lumière est-elle réfléchi ?). C'est à l'issue de la combinaison du travail de modélisation puis de rendu que l'on obtient des images de synthèse telles qu'elles apparaissent notamment dans les films d'animation par ordinateur [Pix].

Remarquons que la classe des objets que l'on modélise dépend de l'application. Ainsi, en conception assistée par ordinateur [Gardan 02], on souhaite interdire la construction de volumes collés entre eux par un point singulier (par exemple, deux cubes collés par un seul sommet commun). En effet, de tels objets ne pourraient pas être usinés. Ainsi, la plupart des *modeleurs géométriques* – parmi lesquels Blender [Ble], Maya [May], ou 3D Studio Max [3ds] – sont basés sur une structure de données qui détermine le type des objets que l'on peut manipuler. Dans la suite, nous parlons de *modèles géométriques* – ou plus simplement *modèles* – pour désigner cette structure de données. Certains modèles comme les *cartes généralisées* [Lienhardt 89, Lienhardt 94] permettent de représenter des objets de dimension quelconque. Néanmoins, la grande majorité des modèles ont été définis afin de manipuler des objets en deux ou trois dimensions (respectivement 2D ou 3D). Notons que les objets 4D sont la plupart du temps utilisés pour l'animation d'objets 3D, on parle alors de 3D+temps [Skapin 01b].

Modèle en fils de fer

Le modèle historique utilisé pour la représentation d'objets 3D est le graphe, on parle parfois de *modèle en fils de fer* (ou *wireframe*). Ici, les objets sont représentés par un ensemble de points de l'espace connectés par des arêtes droites. La principale limitation de ce premier modèle – cause principale de son abandon – réside dans l'ambiguïté des objets manipulés. Par exemple, le cube de la Fig.1.1(b) et le tube de la Fig.1.1(c) peuvent être représentés par la même représentation en fils de fer, celle illustrée en Fig. 1.1(a). En effet, le modèle en fils de fer ne permet pas de distinguer la présence ou l'absence d'une face [Requicha 82].

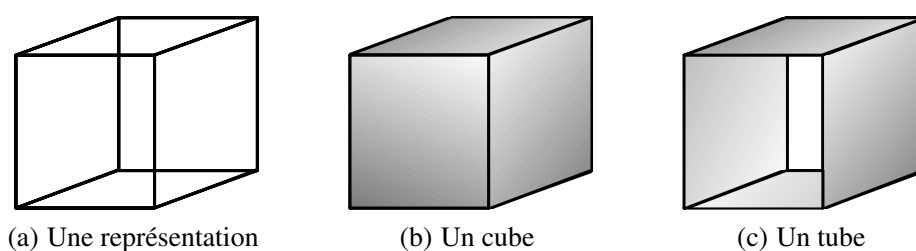


FIGURE 1.1 – Ambiguïté du modèle en fils de fer

Modèle CSG

En *géométrie constructive de solides* [Laidlaw 86] (ou *CSG* pour *Constructive Solid Geometry*), on représente la construction d'un objet par un arbre d'appel d'opérations booléennes. Le principe est le suivant : l'utilisateur part d'un jeu de solides élémentaires qui sont assemblés progressivement par des opérations ensemblistes (union, intersection, etc.). Ainsi, la représentation d'un objet est la donnée d'un arbre – on parle « d'arbre CSG » – qui dresse l'historique des opérations qui mènent à la construction de l'objet. Par exemple, l'objet géométrique 3D de la Fig. 1.2(a) peut être modélisé par l'arbre CSG de la Fig. 1.2(b). On remarque que les feuilles de l'arbre contiennent des solides élémentaires tandis que les nœuds contiennent des opérateurs ensemblistes d'union (\cup) et de soustraction (\setminus). Ce type de modélisation par historique est souvent utilisé dans le cadre du prototypage rapide. Il permet en outre de réutiliser facilement des parties d'objets déjà modélisées afin d'en représenter des variantes (il suffit de modifier localement l'arbre de construction puis de rejouer l'ensemble des opérations). Notons que la puissance de la modélisation par arbre CSG est directement liée à la variété des solides élémentaires et opérateurs disponibles (des opérateurs de translation ou de rotation peuvent être ajoutés). Enfin, dans ce modèle principalement dédié à la conception assistée par ordinateur, la non unicité de représentation des objets – de nombreux arbres permettent de modéliser un même objet – ainsi que la nature implicite de la représentation – l'objet final n'existe pas en tant que tel – rendent difficile la définition d'opérations de manipulation d'objets.

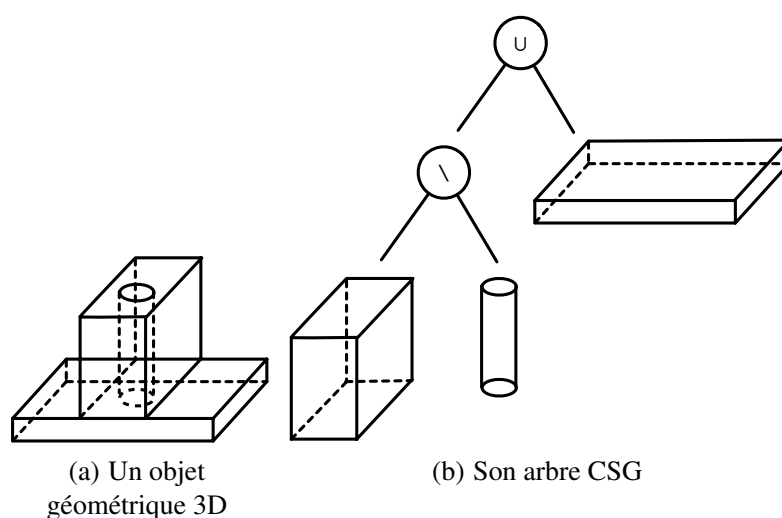


FIGURE 1.2 – Une représentation par arbre CSG

1.2 Modélisation géométrique à base topologique

Les modèles de *représentation par les bords* [Baumgart 75, Weiler 88] (ou modèles *B-rep*) s'opposent aux modèles à historique de construction – comme le modèle CSG – en ce sens qu'ils proposent une représentation explicite des objets. Ici, un objet est décrit par son bord, c'est-à-dire la surface fermée et orientable qui l'enveloppe. Cette surface est elle-même décrite par un ensemble de faces, arêtes et sommets. Les modèles de représentation par les bords les plus communément utilisés de nos jours sont les *modèles cellulaires*, ils s'appuient sur des notions de topologie combinatoire. Ils permettent de représenter des subdivisions d'espaces, c'est-à-dire des objets partitionnés en *cellules* de différentes dimensions : sommets, arêtes, faces et volumes. Un exemple de subdivision 2D est donné en Fig. 1.3. La subdivision est composée de cinq sommets (dénotés par les lettres A à E) connectés par six arêtes (dénotés par les lettres a à f), qui forment deux faces F_1 et F_2 .

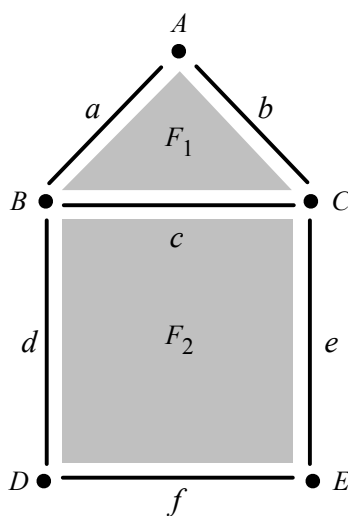


FIGURE 1.3 – Une subdivision 2D

Les modèles cellulaires se situent dans le cadre de la *modélisation géométrique à base topologique* (ou *modélisation topologique*). Ici et dans la suite, nous opposerons très fortement la notion de géométrie (qui relève de la forme et de la position des objets) à la notion de topologie. En modélisation géométrique à base topologique, nous nous focalisons sur la structure des objets, c'est-à-dire leur décomposition en cellules de différentes dimensions. Par exemple, en 3D, nous décomposons les objets en volumes (cellules de dimension 3, ou 3-cellules), faces (2-cellules), arêtes (1-cellules) et sommets (0-cellules). De plus, les modèles topologiques permettent de décrire les relations de voisinage qui existent entre les différentes cellules d'un objet. Nous distinguons deux types de relation de voisinage :

- les liens d'adjacence qui associent deux cellules de même dimension voisines l'une de l'autre. Sur la Fig. 1.3, la face F_1 est adjacente à la face F_2 (et *vice-versa*) ;
- les liens d'incidence qui associent deux cellules de dimension distinctes et voisines l'une de l'autre. Par exemple, le sommet A est incident à l'arête a (et *vice-versa*). Notons que les liens d'incidence permettent de décrire les relations de bords. Dans la pratique, lorsque deux cellules sont incidentes l'une de l'autre, on dit que la cellule de plus faible dimension appartient au bord de l'autre. Par exemple, le sommet A est dans le bord de l'arête a .

Bien que les modèles topologiques représentent principalement la structure des objets, une géométrie peut être associée aux différentes cellules par l'intermédiaire d'un *modèle de plongement*, ou plus simplement d'un *plongement*. Dans la Fig 1.4, deux plongements différents sont associés à une même structure. Dans les deux cas, la structure est un volume comportant notamment trois faces F_1 , F_2 et F_3 .

Le plongement polyédrique de la Fig. 1.4(a) permet de définir des arêtes droites et des faces planes tandis que le plongement arrondi – *via* des courbes et surfaces – de la Fig. 1.4(b) permet de définir des objets aux formes lisses.

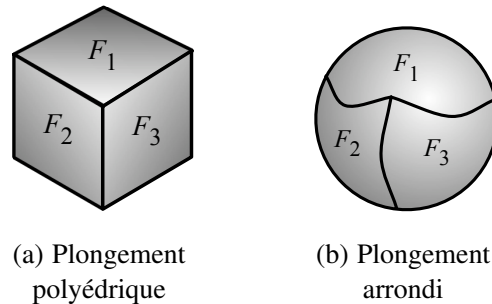


FIGURE 1.4 – Deux types de plongement pour une même structure

Deux arguments permettent de s'orienter vers le choix d'un modèle topologique plutôt que vers le choix d'un modèle purement géométrique, c'est-à-dire où la structure reste implicite. D'une part, la modélisation topologique est naturellement indiquée lorsque la structure des objets manipulés ainsi que les relations de voisinage au sein de cette structure sont déterminantes pour l'application. C'est le cas en modélisation de bâtiments architecturaux où l'on souhaite déterminer, par exemple, quelles pièces sont séparées par une porte donnée [Horna 09]. C'est également le cas en modélisation de systèmes complexes biologiques où les relations de voisinage entre les différents compartiments – par exemple les cellules, organes ou compartiments – déterminent les réactions biochimiques. Notamment, deux protéines ne peuvent être transportées d'un compartiment à un autre que si ces deux compartiments sont voisins. D'autre part, la modélisation topologique permet dans une certaine mesure de s'abstraire de la géométrie afin de simplifier les traitements. En effet, les calculs géométriques – où l'on est régulièrement confronté à des problèmes de précision numérique – sont souvent plus lourds que les traitements effectués sur la seule structure.

De nombreux modèles cellulaires ont été définis. On distingue souvent plusieurs classes de modèles selon qu'ils permettent de représenter des cellules quelconques ou régulières. Les *modèles simpliciaux* comptent parmi les modèles cellulaires où l'on manipule uniquement des cellules régulières (des « triangles » en toute dimension : points, segments, triangles, tétraèdres, *etc.*). Notons que la régularité des cellules trouve son intérêt dans les algorithmes de synthèse d'images qui mettent en jeu les processeurs graphiques¹ [Fernando 04, Pharr 05]. Dans la suite, nous mettons de côté les modèles simpliciaux pour privilégier des modèles plus lâches pour lesquels il n'existe pas de restriction sur la nature des cellules. On parle alors de cellules quelconques. Ces *modèles cellulaires généraux* – par opposition aux modèles simpliciaux – sont souvent classés en fonction du type d'objets qu'ils permettent de représenter. Par exemple, nous pouvons distinguer les modèles cellulaires généraux selon qu'ils permettent ou non de représenter la notion de multi-incidence. Par exemple, sur la Fig. 1.5, l'arête *a* est incidente deux fois au sommet *A* et forme ainsi une boucle.

Parmi les modèles cellulaires généraux, les graphes d'incidence permettent de représenter les objets sous la forme d'un graphe orienté qui capture les relations de voisinage. Un exemple de graphe d'incidence est donné en Fig. 1.6, il représente la subdivision 2D de la Fig. 1.3. À chaque sommet du graphe est associée une cellule de l'objet modélisé et aux arêtes du graphe correspondent les relations de bord entre des *i*-cellules et des (*i* – 1)-cellules. Le modèle des graphes d'incidence ne permet pas de représenter la multi-incidence. Ainsi, on interdit la représentation d'objets tels que celui illustré en Fig. 1.5.

1. Dans la pratique, le traitement d'objets 3D avec facettes triangulaires est intégré à la chaîne de traitement des cartes graphiques.

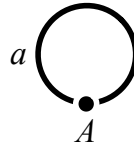


FIGURE 1.5 – Une arête incidente deux fois à un même sommet

En revanche, les graphes d'incidence permettent de représenter des objets incohérents où une 1-cellule est connectée à plus de deux 0-cellules.

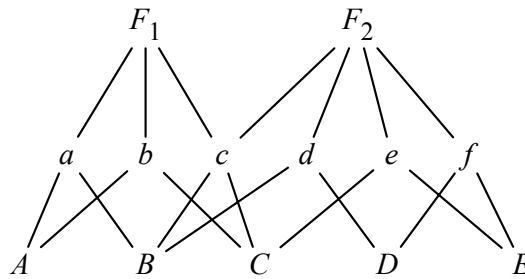


FIGURE 1.6 – Un graphe d'incidence

Une autre limitation des graphes d'incidence est l'ambiguïté du mode de représentation, celle-ci est illustrée en Fig. 1.7. Intuitivement, les objets O_1 et O_2 présentés respectivement en Fig. 1.7(a) et Fig. 1.7(b) se distinguent par leur structure mais semblent posséder la même représentation (voir Fig. 1.7(c)). Cette ambiguïté place les graphes d'incidence dans la catégorie des *modèles non ordonnés*. Dans ces modèles, l'ordre des cellules les unes par rapport aux autres n'apparaît pas. Par exemple, si l'on parcourt la face F_1 dans le sens trigonométrique dans l'objet O_1 puis dans l'objet O_2 , on obtient deux suites d'arêtes différentes qui sont respectivement : (a, c, e, d, f, b) et (a, c, f, d, e, b) . Dans le graphe de la Fig. 1.7(c), cette nuance n'apparaît pas. Rigoureusement, le graphe de la Fig. 1.7(c) ne permet de représenter ni O_1 ni O_2 , puisque l'un comme l'autre présentent un cas de multi-incidence : la face F_1 est incidente deux fois aux sommet C .

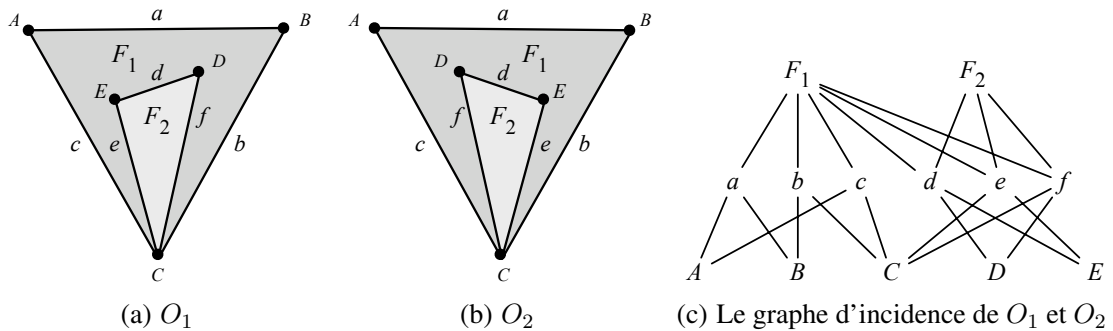


FIGURE 1.7 – Ambiguïté des graphes d'incidence

Dans ce document, nous choisissons de nous placer dans le cadre des *modèles ordonnés*. Ces modèles sont plus fonctionnels dans la plupart des algorithmes basés sur le parcours de structures. Ils sont basés sur des éléments de base abstraits qui correspondent intuitivement à un sommet de l'objet, « vu » d'une

arête incidente à ce sommet, « vue » elle-même d'une face incidente à l'arête, *etc.* Ainsi, les cellules sont représentées implicitement par un assemblage d'éléments abstraits. Le principal intérêt de ce type de modèle est qu'il devient facile de définir précisément l'ensemble des objets que l'on souhaite représenter. Pour cela, il suffit d'ajouter des contraintes dites de *cohérence*. Par exemple, ces contraintes peuvent interdire les arêtes incidentes à plus de deux sommets. Remarquons enfin que les modèles ordonnés permettent de représenter la multi-incidence.

1.3 Cartes généralisées

Le modèle des *cartes généralisées* [Lienhardt 89, Lienhardt 94] ou *G-cartes*, sur lequel nous nous appuyons tout au long de ce manuscrit, fait partie des modèles ordonnés. Les cartes généralisées sont issues de la notion de cartes combinatoires [Tutte 84, Bryant 85]. Elles présentent l'avantage d'être définies mathématiquement, et de manière homogène, en toute dimension. Ainsi, les traitements appliqués aux cartes généralisées – notamment les opérations de construction ou d'édition – peuvent être aussi définis mathématiquement et de manière homogène en toute dimension.

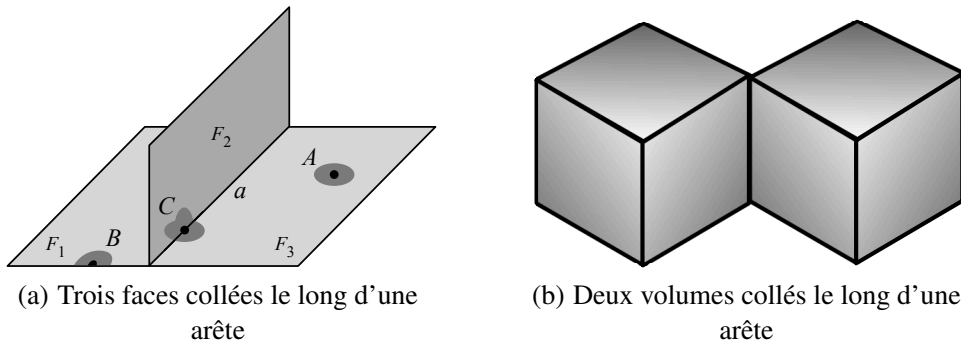


FIGURE 1.8 – Deux objets qui ne sont pas des variétés

Les cartes généralisées permettent de représenter la topologie des subdivisions d'espace qui sont des *quasi-variétés topologiques*, ou plus simplement *quasi-variétés*. Cette famille est proche de la famille des *variétés topologiques* [Agoston 76], ou *variétés*. Intuitivement, une variété de dimension n est un objet de dimension n dont chacun des points possède un voisinage homéomorphe à une boule de dimension n , c'est-à-dire un voisinage que l'on peut « déformer » en une boule de dimension n . Sur la Fig. 1.8(a), le voisinage du point A est homéomorphe à un disque, c'est-à-dire à une boule de dimension 2. Le voisinage du point B est également homéomorphe à un disque. En revanche, le voisinage du point C est homéomorphe à un « feuillet » – c'est-à-dire un ensemble de plans qui s'intersectent – mais pas à un disque. Par conséquent, l'objet formé des faces F_1 , F_2 et F_3 n'est pas une variété de dimension 2. De la même manière, l'objet 3D de la Fig. 1.8(b) constitué de deux volumes collés le long d'une arête n'est pas une variété de dimension 3. En effet, les points de l'arête de contact entre les deux volumes ne possèdent pas de voisinage homéomorphe à une boule de dimension 3 (ils possèdent en fait un voisinage homéomorphe à une demi-boule de dimension 3).

Initialement, la notion de quasi-variété a été introduite pour décrire l'ensemble des objets que l'on peut représenter avec une carte généralisée. La notion de quasi-variété est moins restrictive que la notion de variété. Une conséquence est que toute variété est une quasi-variété (mais la réciproque n'est pas vraie). Intuitivement, une quasi-variété de dimension n peut être définie par un ensemble de n -cellules, collées entre elles le long de $(n-1)$ -cellules. De cette manière, une $(n-1)$ -cellule est incidente à au plus deux n -cellules. L'objet illustré en Fig. 1.9 est une quasi-variété de dimension 3. En effet, les différents volumes sont tous collés le long d'une face. Les faces séparent ainsi au plus deux volumes. En revanche, ce même objet n'est pas une 3-variété. En effet, le point A ne possède pas de voisinage homéomorphe à une boule de dimension 3. Notons enfin que les objets représentés en Fig. 1.8(a) et en Fig. 1.8(b) ne sont

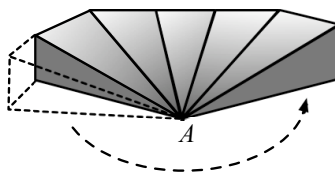
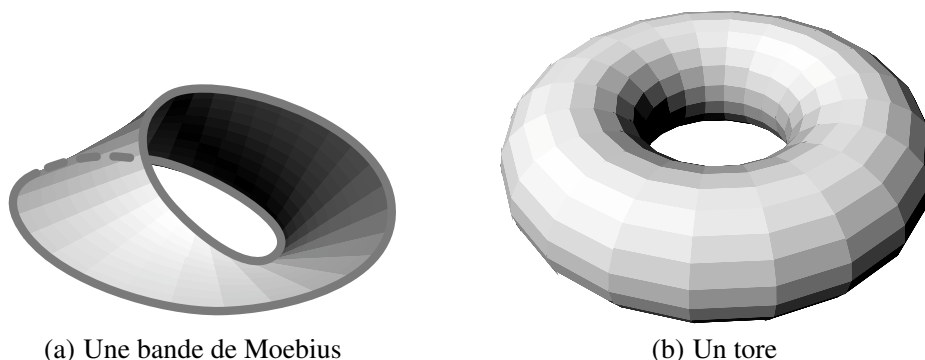


FIGURE 1.9 – Une quasi-variété qui n'est pas une variété

pas, respectivement, des quasi-variétés de dimension 2 et 3. En effet, l'objet de la Fig. 1.8(a) présente une arête incidente à trois faces tandis que l'objet de la Fig. 1.8(b) présente deux volumes collés le long d'une arête, et non d'une face.

Enfin, les cartes généralisées permettent de représenter la topologie d'objets avec ou sans *bords* et *orientables* ou non. Intuitivement, un objet de dimension n est considéré comme orientable si la surface de dimension $(n - 1)$ qui représente son bord sépare l'espace en deux parties distinctes : l'intérieur et l'extérieur de l'objet. La bande de Möbius de la Fig. 1.10(a) – qui est un objet de dimension 2 – est une surface non orientable avec bord. Son bord est constitué de l'arête grise épaisse de la figure. Celle-ci ne sépare pas l'espace de dimension 2 en deux parties distinctes. Le tore de la Fig. 1.10(b) est une surface orientable sans bord.



(a) Une bande de Moebius

(b) Un tore

FIGURE 1.10 – Illustration des notions de bord et d'orientation

1.3.1 Un premier exemple

Intuitivement, le principe de la représentation par G-carte consiste à décomposer l'objet que l'on souhaite représenter en un ensemble d'éléments abstraits élémentaires – appelés *brins* – connectés entre eux. La connexion des brins permet de représenter les cellules topologiques de l'objet.

Prenons l'exemple de l'objet 3D illustré en Fig. 1.11(a). Notons que cet objet est une quasi-variété topologique de dimension 3. En effet, il est composé de deux volumes (ou 3-cellules) collés le long d'une face (ou 2-cellule). Les Fig. 1.11(b) à 1.11(e) décrivent les quatre étapes de décomposition de l'objet :

- en Fig. 1.11(b), l'objet est décomposé en deux volumes. Remarquons que les surfaces – ou enveloppes 2D – des deux volumes sont des quasi-variétés de dimension 2 (elles sont également des variétés de dimension 2) ;
- en Fig. 1.11(c), chaque volume est décomposé en un ensemble de faces déconnectées. Pour plus de clarté, seules les parties visibles de l'objet sont illustrées. On remarque que le bord de chaque face est une variété de dimension 1. En effet, chaque sommet relie exactement deux arêtes. Notons

également que les faces n'ont pas de structure prédéfinie, certaines sont triangulaires et d'autres quadrangulaires ;

- en Fig. 1.11(d), chaque face est décomposée en un ensemble d'arêtes ;
- en Fig. 1.11(e), les arêtes sont décomposées en brins. Ainsi, on remarque que les brins peuvent être vus intuitivement comme des demi-arêtes.

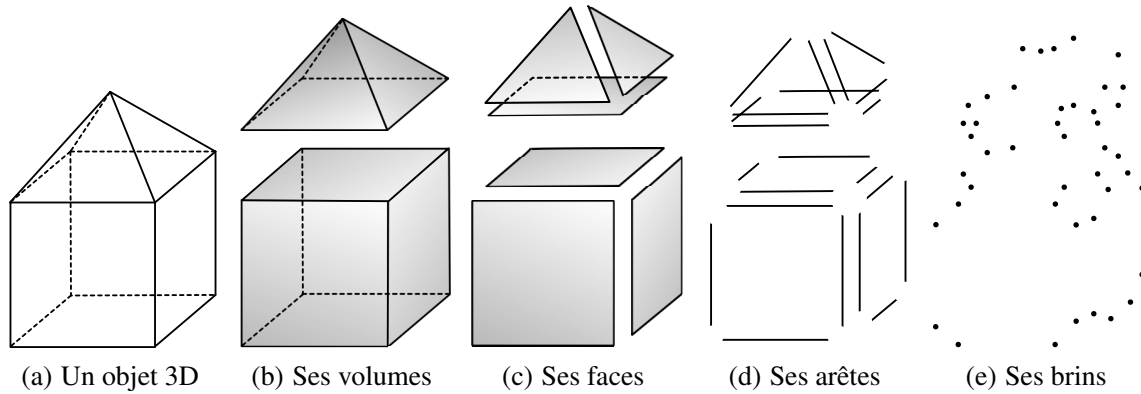


FIGURE 1.11 – Décomposition d'un objet 3D en volumes, faces, arêtes et brins

La seule décomposition de l'objet de départ (Fig. 1.11(a)) en un ensemble de brins (Fig. 1.11(e)) ne permet plus de retrouver les différentes cellules topologiques qui composent l'objet. Volumes, faces et arêtes ont disparu. Nous avons vu en section 1.2 que les modèles géométriques à base topologique, tels que les cartes généralisées, permettent de manipuler de concert la structure des objets (c'est-à-dire leur décomposition en cellules topologiques) et les relations de voisinages qui lient les éléments issus de la décomposition. Ce sont ces relations de voisinages qui permettent de retrouver les différentes cellules topologiques. Dans une carte généralisée de dimension n , chaque brin possède $n + 1$ relations de voisinage numérotées de α_0 à α_n . La sémantique de chacune de ces liaisons est illustrée dans les Fig. 1.12(a) à 1.12(d). Rappelons que seules les parties visibles dans l'objet de la Fig. 1.11(a) sont représentées :

- en Fig. 1.12(a), les liaisons α_0 connectent les brins et modélisent ainsi les 1-cellules de l'objet (voir Fig. 1.11(d)) ;
- en Fig. 1.12(b), les liaisons α_1 connectent les 1-cellules pour former les 2-cellules (voir Fig. 1.11(c)) ;
- en Fig. 1.12(c), les liaisons α_2 connectent les 2-cellules et modélisent ainsi les 3-cellules (voir Fig. 1.11(b)) ;
- en Fig. 1.12(d), les liaisons α_3 connectent les 3-cellules. On retrouve alors la topologie de l'objet initial (voir Fig. 1.11(a)), à l'exception des faces cachées qui ne sont pas représentées ici.

Ainsi, on remarque que dans une carte généralisée, les i -cellules sont représentées implicitement par des assemblages de brins. De plus, l'exemple de la Fig. 1.12 montre qu'une liaison α_i (pour $i \in [0; n]$) permet de connecter deux i -cellules entre elles.

Dans la suite de ce chapitre, nous introduisons la notion de cartes généralisées. Pour cela, nous commençons par introduire une structure intermédiaire : les pré-G-cartes. Intuitivement, une pré-G-carte est une structure composée de brins qui comme dans la Fig. 1.12(d), sont munis de toutes leurs relations de voisinage. Dans un second temps, nous définissons les cartes généralisées comme étant des pré-G-cartes munies d'une contrainte de cohérence particulière qui assure que les objets modélisés sont des quasi-variétés.

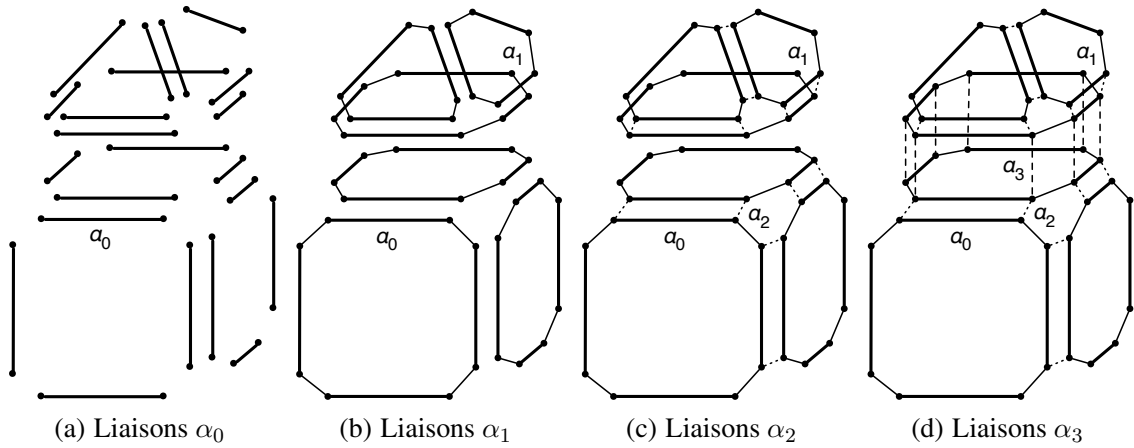


FIGURE 1.12 – Ajout des relations de voisinages entre les brins de la Fig. 1.11(e)

1.3.2 Pré-G-cartes

En introduction, nous avons présenté notre souhait d'animer nos structures topologiques en utilisant le formalisme de la transformation de graphes. Dans cette section, nous définissons les pré-G-cartes comme des cas particuliers de graphes partiellement étiquetés. Ainsi, notre approche se distingue de l'approche classique où les G-cartes sont définies de manière algébrique [Lienhardt 89, Lienhardt 94]. Remarquons cependant que les deux approches sont strictement équivalentes.

Rappelons qu'un graphe partiellement étiqueté G (voir en annexe A.2.1, la définition 54) est défini par un sextuplet (V, E, s, t, l_V, l_E) avec :

- V et E respectivement des ensembles de sommets² et d'arcs ;
- s et t les fonctions qui associent à chaque arc un sommet source et un sommet cible. On dit qu'un arc est *incident* à son sommet source ;
- l_V et l_E respectivement les fonctions d'étiquetage des sommets et des arcs. Par convention, l'étiquette \perp dénote les sommets et les arcs qui ne sont pas étiquetés.

De plus, lorsqu'un graphe contient un arc (v, l, v') – c'est-à-dire de sommet source v , de sommet cible v' et d'étiquette l – ainsi que l'arc d'orientation inverse (v', l, v) , on parle d'arête l – ou encore d'arête étiquetée par l – qui connecte les sommet v et v' . Une telle arête est représentée graphiquement par un trait reliant v et v' et décoré par l .

Nous représentons les brins d'une pré-G-carte de dimension n par des sommets de graphe. Les relations de voisinage sont alors modélisées par des arcs de graphe partiellement étiquetés sur $\{\alpha_0, \dots, \alpha_n\}$. Remarquons que dans l'approche algébrique, les relations de voisinages α_0 à α_n sont des involutions, c'est-à-dire des applications α_i (avec $0 \leq i \leq n$) des brins vers les brins telle que α_i^2 est l'identité. Ainsi, dans notre approche basée sur les graphes, chaque brin d'une pré-G-carte de dimension n est le sommet source d'exactly n arcs incidents étiquetés respectivement par α_0 à α_n . Cette contrainte, que nous appelons contrainte d'arcs incidents – puisqu'elle porte sur les arcs incidents aux sommets d'un graphe – peut être définie de la manière suivante.

Définition 1 (contrainte d'arcs incidents). Soient Σ_V un ensemble d'étiquettes de sommets et Σ_E un ensemble d'étiquettes d'arcs. Soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ un graphe partiellement étiqueté sur (Σ_V, Σ_E) et $\Sigma_{E'} \subset \Sigma_E$ un ensemble fini d'étiquettes.

2. Ici et dans la suite, le terme « sommet » désigne les sommets d'un graphe et non des 0-cellules.

Soit $v \in V_G$ un sommet de G . Le sommet v satisfait la contrainte d'arcs incidents $\text{Incede}_{v,G}(\Sigma_{E'})$ si pour toute étiquette $l \in \Sigma_{E'}$, le sommet v est source d'un unique arc incident étiqueté par l , et pour toute étiquette $l' \in \Sigma_E \setminus \Sigma_{E'}$, v est source d'aucun arc étiqueté par l' .

Le graphe G satisfait la contrainte d'arcs incidents $\text{Incede}_G(\Sigma_{E'})$ si pour tout sommet $v \in V_G$, v satisfait $\text{Incede}_{v,G}(\Sigma_{E'})$.

Soit H un graphe partiellement étiqueté sur (Σ_V, Σ_E) , la contrainte d'arcs incidents s'étend naturellement à la structure $G \setminus H$, comme suit. Pour tout sommet v de $G \setminus H$, la contrainte $\text{Incede}_{v,G \setminus H}(\Sigma_{E'})$ est satisfaite si et seulement si pour toute étiquette $l \in \Sigma_{E'}$, v est source d'un unique arc d'étiquette l , et pour toute étiquette $l' \in \Sigma_E \setminus \Sigma_{E'}$, v n'est source d'aucun autre arc étiqueté par l' . La contrainte d'arc incident $\text{Incede}_{G \setminus H}(\Sigma_{E'})$ est satisfaite si et seulement si $\text{Incede}_{v,G \setminus H}(\Sigma_{E'})$ est satisfaite pour tout sommet v de $G \setminus H$.

Remarquons que nous définissons la contrainte d'arcs incidents dans le cadre général des graphes partiellement étiquetés, en dehors du contexte des pré-G-cartes. Nous définissons la notion de pré-G-carte de la manière suivante.

Définition 2 (pré-G-carte). Soient $n \geq 0$, notons $\mathcal{A}_n = \{\alpha_0, \dots, \alpha_n\}$ l'ensemble des étiquettes d'arcs de dimension n . Une pré-G-carte de dimension n est un graphe topologique G de dimension n – c'est-à-dire un graphe partiellement étiqueté sur $(\emptyset, \mathcal{A}_n)$ – tel que :

- G est non orienté (voir en annexe A.2.1, la définition 56);
- la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$ est satisfaite.

Dans notre définition de pré-G-carte, nous n'étiquetons pas les sommets. Dans la pratique, les étiquettes des sommets d'une pré-G-carte – ou d'une G-carte – permettent notamment d'associer un plongement aux objets. Pour l'heure, nous nous focalisons sur les aspects topologiques, qui sont indépendants du plongement.

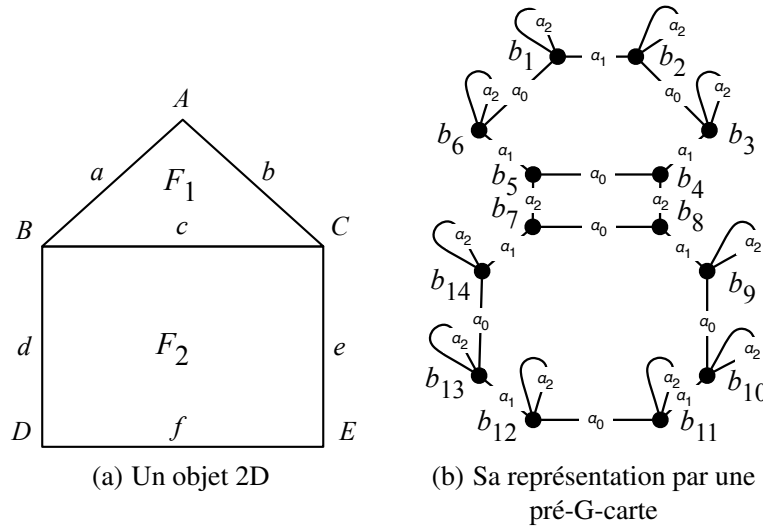


FIGURE 1.13 – Représentation d'un objet 2D par une pré-G-carte de dimension 2

La Fig. 1.13 illustre notre définition de pré-G-carte. On cherche à modéliser la maison en deux dimensions de la Fig. 1.13(a). Sur la Fig. 1.13(b), on remarque deux 2-cellules connectées par α_2 ; elles modélisent la façade et le toit de la maison. Dans la définition 2, on impose qu'une pré-G-carte soit non orientée. Ce choix dérive directement de la notion d'involution, utilisée dans la définition algébrique des

pré-G-cartes. En effet, lorsqu'un brin b est lié à un brin b' par une involution α_i (avec $0 \leq i \leq n$), alors b' est lui-même lié à b par ce même α_i . Ainsi, on remarque que nos liaisons α_i sont toutes représentées par des arêtes sur la Fig. 1.13(b).

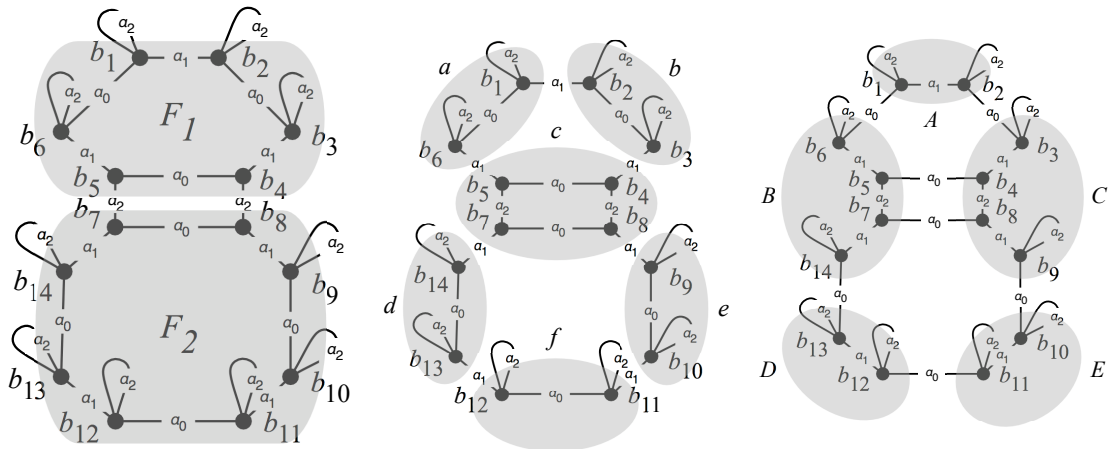
Intéressons-nous à présent à la contrainte d'arcs incidents. On remarque que dans la Fig. 1.13(b), afin que la contrainte d'arcs incidents soit satisfaite par les brins du bord b_1, b_2, b_3, b_6 , et b_9 à b_{14} , on ajoute à chacun de ces brins une boucle incidente étiquetée par α_2 . Nous représentons graphiquement les boucles par des traits plutôt que par des flèches. En effet, une boucle est sa propre inverse, elle peut donc être vue aussi bien comme un arc que comme une arête.

Dans une carte généralisée, les cellules topologiques sont représentées implicitement par des assemblages de brins. La notion d'*orbite* nous permet de définir les cellules. Intuitivement, une orbite est un sous-graphe généré par un sommet et un ensemble d'étiquettes d'arcs. La définition d'orbite ne se limite pas aux pré-G-cartes, elle porte sur les graphes partiellement étiquetés généraux.

Définition 3 (orbite). Soient Σ_V un ensemble d'étiquettes de sommets et Σ_E un ensemble d'étiquettes d'arcs. Soient $G = (V, E, s, t, l_V, l_E)$ un graphe partiellement étiqueté sur (Σ_V, Σ_E) , β un mot de Σ_E^* dont toutes les lettres sont différentes (voir annexe A.1) et v un sommet de V . L'orbite $\langle \beta \rangle (v)$ est le sous-graphe généré par v et Σ_β , avec Σ_β l'ensemble des étiquettes composants le mot β (voir en annexe A.2.3, la définition 61). Une telle orbite est dite incidente à v .

Par définition, une orbite $\langle \beta \rangle (v)$ est étiquetée sur les seules étiquettes d'arcs $\Sigma_\beta \subset \Sigma_E$. Nous appelons $\langle \beta \rangle$ le *type* du graphe $\langle \beta \rangle (v)$. Plus généralement, pour tout mot $\beta \in \Sigma_E^*$, $\langle \beta \rangle$ est l'ensemble des graphes partiellement étiquetés sur (\emptyset, Σ_E) dont toutes les étiquettes d'arcs appartiennent à $\Sigma_\beta \subset \Sigma_E$. Remarquons que pour tous sommets v et v' d'une même orbite $\langle \beta \rangle (v)$, nous avons $\langle \beta \rangle (v) = \langle \beta \rangle (v')$. Nous définissons la notion de cellule, cette fois-ci dans le cadre des pré-G-cartes, de la manière suivante.

Définition 4 (i-cellule). Soient $G = (V, E, s, t, l_V, l_E)$ une pré-G-carte de dimension n , v un sommet de V et β_i (avec $0 \leq i \leq n$) le mot constitué de la suite d'étiquettes α_0 à α_n privée de l'étiquette α_i . La i -cellule incidente à v est l'orbite $\langle \beta_i \rangle (v)$ incidente à v .



(a) 2-cellules de type $\langle \alpha_0 \alpha_1 \rangle$ (b) 1-cellules de type $\langle \alpha_0 \alpha_2 \rangle$ (c) 0-cellules de type $\langle \alpha_1 \alpha_2 \rangle$

FIGURE 1.14 – Les cellules de la maison

Dans la section 1.2, nous avons vu que les modèles ordonnés s'appuient sur un élément de base, dans notre cas le brin, qui correspond intuitivement à un sommet de l'objet modélisé « vu » d'une arête, « vue » elle-même d'une face, « vue » elle-même d'un volume dans le cas d'un objet 3D. Ainsi, on remarque que chaque brin d'une pré-G-carte appartient à exactement une cellule par dimension. Cette observation est illustrée sur la Fig. 1.14. Le brin b_9 appartient à la 2-cellule $\langle \alpha_0 \alpha_1 \rangle (b_9)$ (voir Fig. 1.14(a)) qui modélise la façade F_2 , à la 1-cellule $\langle \alpha_0 \alpha_2 \rangle (b_9)$ (voir Fig. 1.14(b)) qui modélise l'arête géométrique e et à la 0-cellule $\langle \alpha_1 \alpha_2 \rangle (b_9)$ (voir Fig. 1.14(c)) qui modélise le sommet géométrique C .

Prenons quelques autres exemples de cellules illustrées sur la Fig. 1.14 :

- le toit F_1 de la maison est modélisé par la 2-cellule $\langle \alpha_0 \alpha_1 \rangle (b_1)$ incidente au brin b_1 (voir Fig. 1.14(a)). Cette 2-cellule est composée des brins b_1 à b_6 ainsi que des arêtes étiquetées par α_0 et α_1 qui connectent ces brins. Remarquons que le retrait de l'étiquette α_2 au mot associé à une 2-cellule nous assure de ne pas regrouper dans la même 2-cellule le toit et la façade de la maison (qui sont connectés par α_2) ;
- l'arête géométrique a de la maison est modélisée par la 1-cellule $\langle \alpha_0 \alpha_2 \rangle (b_6)$ (voir Fig. 1.14(b)). Elle comprend les brins b_6 et b_1 ainsi que leurs arêtes incidentes étiquetées par α_0 ;
- le sommet géométrique B de la maison est modélisé par la 0-cellule $\langle \alpha_1 \alpha_2 \rangle (b_6)$ (voir Fig. 1.14(c)). Elle comprend les brins b_6 , b_5 , b_7 et b_{14} ainsi que leurs arêtes incidentes étiquetées par α_1 et α_2 .

1.3.3 Contraintes de cycles

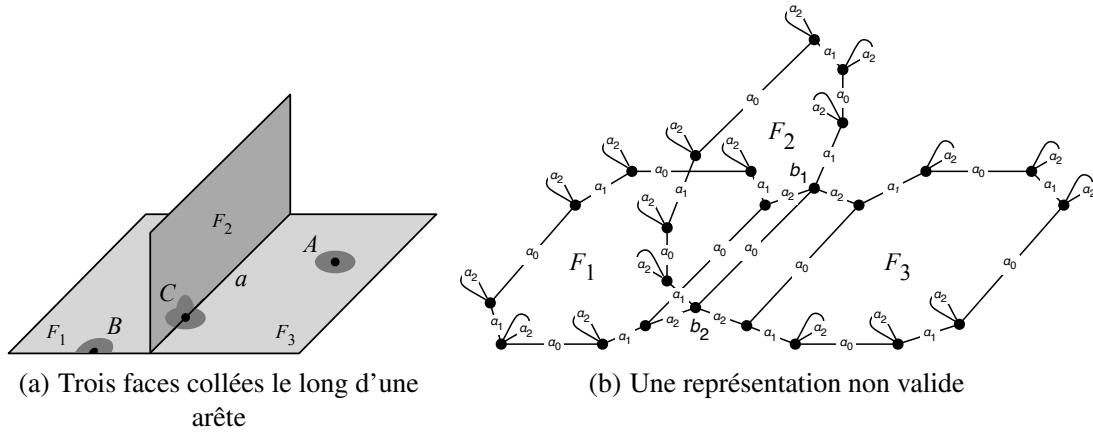


FIGURE 1.15 – Un objet qui ne peut pas être représenté avec une pré-G-carte

Dans cette section, nous définissons les cartes généralisées comme étant des pré-G-cartes munies d'une contrainte de cohérence supplémentaire – nous l'appelons la *contrainte de cycle* – qui nous assurent que les objets modélisés sont des quasi-variétés.

Les cartes généralisées permettent de représenter des subdivisions d'espace qui sont des quasi-variétés. À ce titre, nous avons déjà étudié l'objet de la Fig. 1.15(a) qui viole la condition de quasi-variété de dimension 2. Sur cet objet, trois 2-cellules F_1 , F_2 et F_3 sont collées le long d'une unique 1-cellule a . On remarque en Fig. 1.15(b) que cet objet ne peut pas être représenté à l'aide d'une pré-G-carte de dimension 2. En effet, la contrainte d'arcs incidents n'est pas satisfaite puisque les brins b_1 et b_2 sont les sommets sources de deux arcs incidents étiquetés par α_2 . Ainsi, la contrainte d'arcs incidents des pré-G-cartes permet d'éliminer certains objets qui ne sont pas des quasi-variétés. Plus généralement, cette contrainte interdit les objets où une $(i - 1)$ -cellule borde plus de deux i -cellules.

L'objet 2D illustré en Fig. 1.16(a) n'est pas non plus une quasi-variété de dimension 2. En effet, les 2-cellules F_1 et F_2 sont collées le long d'une 0-cellule. Or, nous savons que deux i -cellules doivent être collées le long d'une $(i-1)$ -cellule. Cependant, ce nouvel objet peut être représenté par une pré-G-carte. La figure 1.16(b) illustre cette représentation, remarquons que la contrainte d'arcs incidents y est satisfaite. Une contrainte de cohérence supplémentaire est donc nécessaire pour rejeter ce type d'objet.

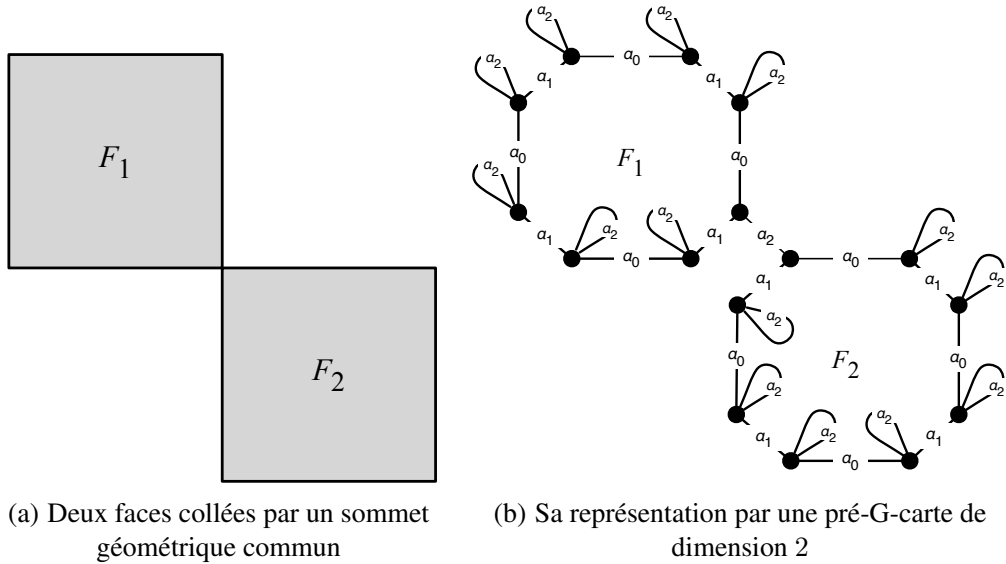


FIGURE 1.16 – Un objet qui n'est pas une quasi-variété de dimension 2 mais qui peut être représenté par une pré-G-carte de dimension 2

Corrigeons l'exemple précédent en collant les deux 2-cellules F_1 et F_2 le long d'une 1-cellule, comme l'impose la condition de quasi-variété de dimension 2. Le résultat est illustré en Fig. 1.17(a). On remarque en Fig. 1.17(b) que les brins b_4, b_3, b_{16} et b_{15} sont organisés en un cycle étiqueté par $\alpha_0\alpha_2\alpha_0\alpha_2$ (voir en annexe A.2.3, la définition 60 de chemin).

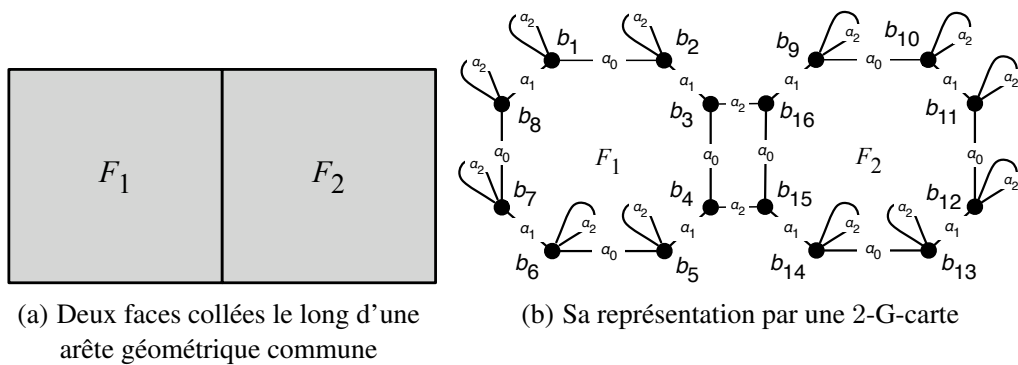


FIGURE 1.17 – Représentation d'un objet 2D par une 2-G-carte

Nous définissons la contrainte de cycles de la manière suivante.

Définition 5 (contrainte de cycles). Soit $n \geq 0$. Soit $G = (V, E, s, t, l_V, l_E)$ un graphe topologique de dimension n (voir la définition 2 page 11 des pré-G-cartes). Le graphe G satisfait la contrainte de cycles $Cycle_G(\mathcal{A}_n)$ si pour toute couple d'étiquettes $(\alpha_i, \alpha_j) \in \Sigma_E^2$ tel que les indices i et j de $[0, n]$ vérifient

$i + 2 \leq j$ et pour tout sommet v de V , il existe un cycle étiqueté par $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v (voir en annexe A.2.3, la définition 60 de chemin).

Nous pouvons maintenant définir la notion de carte généralisée de la manière suivante :

Définition 6 (G-carte). Soit $n \geq 0$. Une carte généralisée de dimension n – ou n -G-carte – est une pré-G-carte $G = (V, E, s, t, l_V, l_E)$ de dimension n qui satisfait la contrainte de cycles $\text{Cycle}_G(\mathcal{A}_n)$.

Puisque par définition, une n -G-carte est une pré-G-carte de dimension n , elle est non orientée et vérifie la contrainte d'arcs incidents. Notons également que dans une n -G-carte, les boucles α_i (pour $0 \leq i \leq n$) incidentes aux brins du *bord* permettent d'assurer la satisfaction de la contrainte de cycles sur tous les brins de la n -G-carte. Par exemple, sur la Fig. 1.17(b), il existe bien un cycle $\alpha_0 \alpha_2 \alpha_0 \alpha_2$ qui part de b_1 . Il contient les brins b_1 et b_2 ainsi que leurs arêtes incidentes α_0 et α_2 .

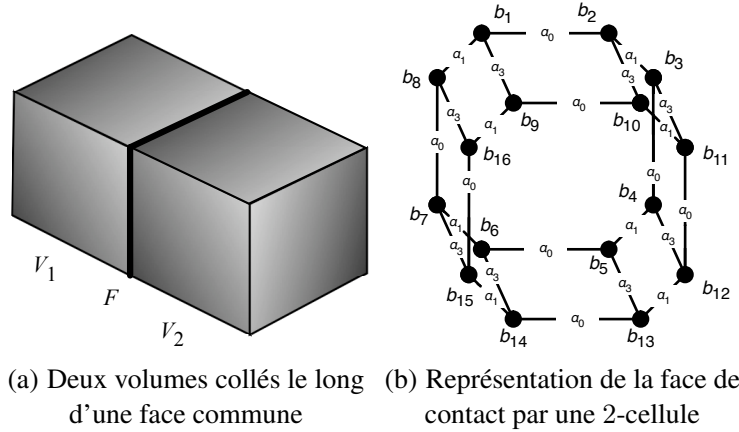


FIGURE 1.18 – Représentation de deux volumes collés le long d'une face

Un exemple en 3D est donné en Fig. 1.18. Sur la Fig. 1.18(a), deux 3-cellules V_1 et V_2 sont collées le long d'une 2-cellule F (représentée en gras sur la figure). Cette 2-cellule est représentée sur la Fig. 1.18(b). Les brins b_1 à b_8 appartiennent à la 3-cellule V_1 et les brins b_9 à b_{16} appartiennent à la 3-cellule V_2 . Remarquons, que cette représentation n'est pas une 3-G-carte (les liens α_2 n'apparaissent pas) mais une orbite face F de type $\langle \alpha_0 \alpha_1 \alpha_3 \rangle$, c'est-à-dire une 2-cellule. Dans une 3-G-carte, la contrainte de cycles se traduit par la présence de cycles $\alpha_0 \alpha_2 \alpha_0 \alpha_2$, $\alpha_0 \alpha_3 \alpha_0 \alpha_3$ et $\alpha_1 \alpha_3 \alpha_1 \alpha_3$. Comme F ne contient pas d'arc étiqueté par α_2 , seuls les deux derniers cycles s'appliquent à F . En particulier, sur la Fig. 1.18(b), la présence de tels cycles impose que les brins b_1 à b_8 soient reliés deux à deux aux brins b_9 à b_{16} par α_3 . La condition de quasi-variété est ainsi satisfaite : les deux 3-cellules V_1 et V_2 sont collées le long de la 2-cellule F .

1.4 Opérations de base

Les opérations de la modélisation géométrique permettent de manipuler les objets, c'est-à-dire de leur appliquer des transformations. Nous distinguons les opérations de base des opérations dites de haut niveau. Dans le cas des G-cartes, les opérations de base permettent de construire et de déconstruire les objets. Elles permettent d'ajouter et de retirer des brins et des liens de voisinage entre les brins. À la différence de ces opérations de base qui modifient localement les objets, les opérations de haut niveau

mettent souvent en jeu de nombreuses cellules topologiques. Elles proposent des modifications plus globales des objets.

De nombreuses opérations ont été définies mathématiquement sur les cartes généralisées. Dans cette section, nous présentons brièvement les quatre opérations de base des cartes généralisées : l'*ajout et la suppression d'un brin* puis la *couture* et la *décousure* (qui permettent de modifier les liens de voisinage entre les cellules topologiques).

Dans cette section, de même que dans la section suivante où nous aborderons les opérations de haut niveau, nous traitons des opérations purement topologiques. Ces opérations s'appliquent à des graphes topologiques, c'est-à-dire des graphes dépourvus d'étiquettes de sommets.

1.4.1 Ajout d'un brin

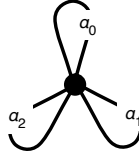


FIGURE 1.19 – Un brin de 2-G-carte isolé

La plus simple des opérations que l'on peut appliquer à une carte généralisée consiste à lui ajouter un nouveau brin. Un tel brin est dit isolé en ce sens qu'au moment de son ajout, il n'est connecté à aucun autre brin. Ainsi, tous ses arcs incidents sont des boucles. Un exemple de brin isolé dans le cas d'une 2-G-carte est donné en Fig. 1.19. Cette première opération de base, très simple, est définie mathématiquement de la manière suivante :

Définition 7 (ajout d'un brin). Soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ une n -G-carte sur et un brin b tel que $b \notin V_G$. La n -G-carte $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$, résultat de l'ajout du brin b dans G , est définie par :

- $V_H = V_G \cup \{b\}$;
- $E_H = E_G \cup \{(b, \alpha_i, b) \mid 0 \leq i \leq n\}$;
- s_H, t_H et l_{E_H} sont définis canoniquement au regard de E_H (voir en annexe A.2.1, l'exemple de la page 208) ;
- pour tout $v \in V_H$, $l_{V_H}(v) = \perp$.

1.4.2 Suppression d'un brin

Comme dans l'opération d'ajout d'un brin, l'opération de suppression ne manipule que des brins isolés. En effet, supprimer un brin b connecté à un brin voisin (voir Fig. 1.20(a)) ferait apparaître une arête pendante (arête α_1 de la Fig. 1.20(b)), c'est-à-dire une arête qui n'est pas une boucle et qui est incidente à un unique brin. Pour éviter cela, la suppression d'un brin nécessite sa déconnexion préalable. Cette seconde opération peut être définie mathématiquement de la manière suivante :

Définition 8 (suppression d'un sommet). Soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ une n -G-carte et un brin $b \in V_G$ tel que tout arc de source b est une boucle. La n -G-carte $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$, résultat de la suppression du brin b dans G , est définie par :

- $V_H = V_G - \{b\}$;

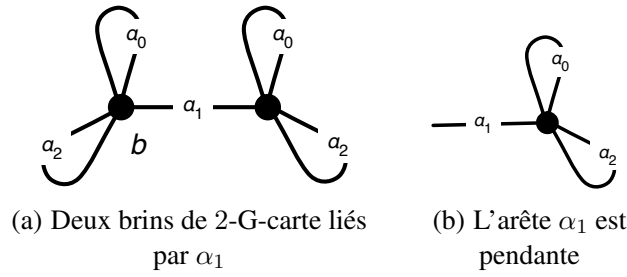


FIGURE 1.20 – Création d'une arête pendante suite à la suppression d'un brin et de ses boucles α_0 et α_1 incidentes

- $E_H = E_G - \{e \mid s_G(e) = b \text{ ou } t_G(e) = b\}$;
- s_H, t_H et l_{E_H} sont respectivement les restrictions de s_G, t_G et l_{E_G} sur E_H ;
- pour tout $v \in V_H, l_{V_H}(v) = \perp$.

1.4.3 Couture

L'opération de couture – nous l'appelons en fait la i -couture – permet d'établir des connexions entre les brins d'une carte généralisée. Elle permet en fait de coller deux i -cellules le long d'une $(i-1)$ -cellule, respectant ainsi le cadre des quasi-variétés.

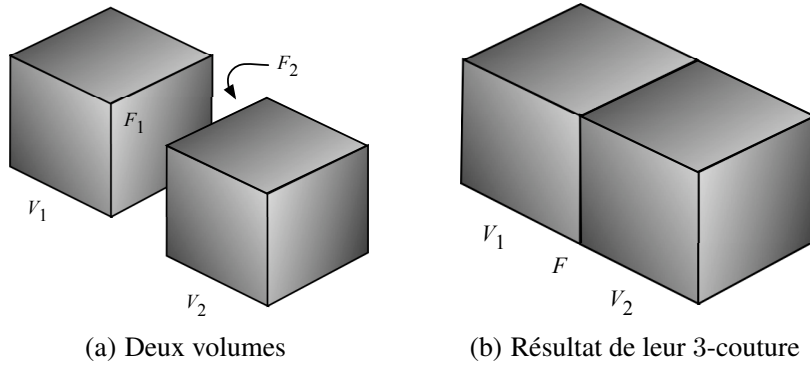


FIGURE 1.21 – 3-couture de deux volumes le long d'une face

Une illustration de la 3-couture de deux 3-cellules le long d'une 2-cellule est donnée en Fig. 1.21. Sur la Fig. 1.21(a), deux volumes V_1 et V_2 sont représentés. On identifie deux faces particulières F_1 et F_2 le long desquelles seront collés V_1 et V_2 . Ainsi, en Fig. 1.21(b), résultat de l'opération de 3-couture, les faces F_1 et F_2 sont réunies en une unique face F de contact entre V_1 et V_2 .

Sur la Fig. 1.22, on remarque qu'il peut exister plusieurs manières de 3-coudre deux objets. Dans la Fig. 1.22(a), deux volumes V_1 et V_2 sont représentés. On note sur les Fig. 1.22(b) et 1.22(c), que le résultat de la 3-couture diffère selon que l'on colle les faces F_1 et F_2 après « translation » (voir Fig. 1.22(b)) ou après « rotation » de V_2 (voir Fig. 1.22(c)).

Afin de contrôler finement les mécanismes de la couture, plaçons-nous à l'échelle des brins d'une G-carte. Reprenons les exemples précédents (voir Fig. 1.21) où il s'agissait de 3-coudre deux volumes V_1 et V_2 le long de deux faces (carrées) F_1 et F_2 . La face F_1 est représentée par la 2-cellule $\langle \alpha_0 \alpha_1 \rangle (b_1)$ dessinée en traits gras sur la Fig. 1.23. De la même manière, la face F_2 est représentée par la 2-cellule $\langle \alpha_0 \alpha_1 \rangle (b_9)$ dessinée en traits fins sur cette même figure. Notons que dans la mesure où nous consi-

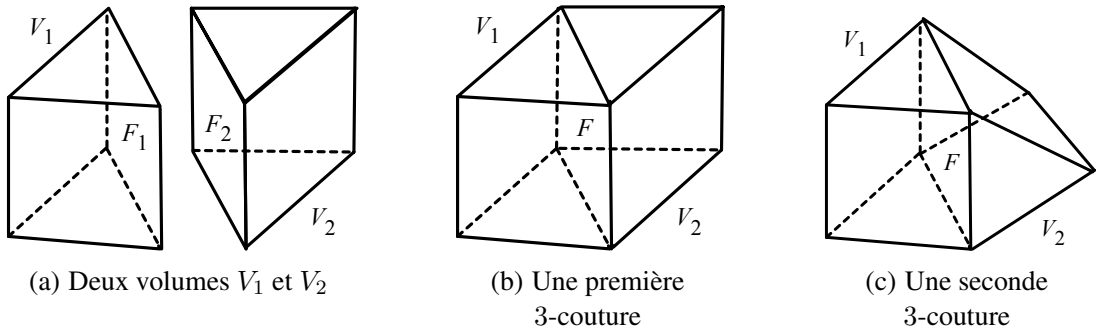


FIGURE 1.22 – Plusieurs manières de 3-coudre deux volumes

dérons des 2-cellules, aucune liaison α_2 n'apparaît (voir section 1.3.2, définition 4).

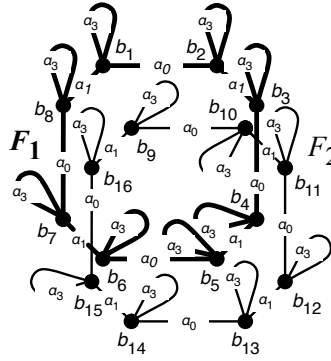


FIGURE 1.23 – Deux 2-cellules

Afin de 3-coudre les volumes V_1 et V_2 le long de F_1 et F_2 , nous devons établir des liaisons α_3 entre les brins de F_1 et les brins de F_2 . Choisissons deux brins b_1 et b_9 et connectons-les par α_3 (voir Fig. 1.24(a)). On remarque que l'ajout de cette seule liaison entre les brins b_1 et b_9 produit un objet non cohérent, puisqu'il ne satisfait pas la contrainte de cycles des 3-G-cartes (voir définition 5 page 14). En effet, les brins b_1, b_2, b_{10} et b_9 n'apparaissent pas dans un cycle étiqueté par $\alpha_0\alpha_3\alpha_0\alpha_3$. De la même manière les brins b_1, b_8, b_{16} et b_9 n'apparaissent pas dans un cycle $\alpha_1\alpha_3\alpha_1\alpha_3$. Afin de rétablir la cohérence pour ces brins, d'autres liens α_3 doivent être ajoutés.

Sur la Fig. 1.24(b), nous ajoutons des liens α_3 entre, respectivement, les brins b_2 et b_{10} et les brins b_8 et b_{16} . De cette manière, il existe un cycle $\alpha_0\alpha_3\alpha_0\alpha_3$ qui passe par les brins b_1, b_2, b_{10} et b_9 . Les brins b_1, b_8, b_{16} et b_9 sont également organisés en un cycle $\alpha_1\alpha_3\alpha_1\alpha_3$. Cependant, l'ajout d'un lien α_3 entre les brins b_2 et b_{10} entraîne que ces deux brins, de même que les brins b_3 et b_{11} , n'apparaissent plus dans un cycle $\alpha_1\alpha_3\alpha_1\alpha_3$. De la même manière, les brins b_8, b_{16}, b_{15} et b_{17} n'apparaissent plus dans un cycle $\alpha_0\alpha_3\alpha_0\alpha_3$. Le problème d'incohérence est donc déplacé.

Afin de rétablir la contrainte de cycles pour tous les brins des 2-cellules F_1 et F_2 , il convient d'ajouter une liaison α_3 entre, respectivement, les couples de brins b_7 et b_{15} , b_6 et b_{14} , b_5 et b_{13} , b_4 et b_{12} , et enfin entre b_3 et b_{11} . Ainsi, en Fig. 1.25, les brins des 2-cellules F_1 et F_2 sont tous liés deux à deux et l'opération de 3-couture des volumes V_1 et V_2 est finalisée. Notons que sur cette même figure, les 2-cellules initiales F_1 et F_2 sont réunies en une même 2-cellule correspondant à la face de contact F de type $\langle \alpha_0\alpha_1\alpha_3 \rangle$ de la Fig. 1.21(b).

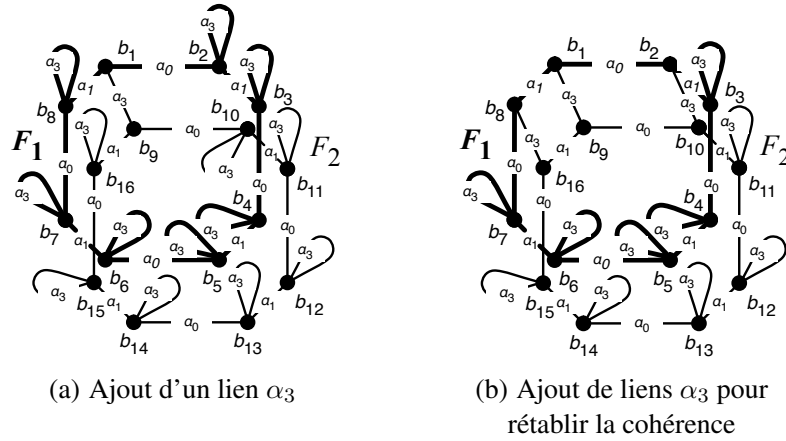
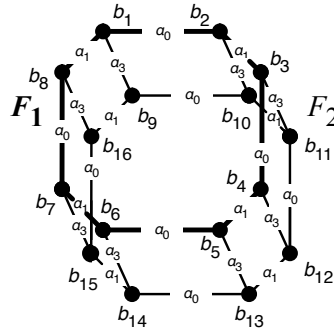


FIGURE 1.24 – Rétablissement de la contrainte de cycles

FIGURE 1.25 – Connexion par α_3 des 2-cellules F_1 et F_2

On remarque en Fig. 1.23 que la 3-couture de F_1 et F_2 est possible dans la mesure où ces 2-cellules sont isomorphes. En effet, elles possèdent exactement le même nombre de brins reliés par les mêmes liaisons. La Fig. 1.26 illustre la notion d'isomorphisme de graphes. Sur la Fig. 1.26(a), les deux orbites $O_1 = \langle \alpha_0 \alpha_1 \rangle (b)$ (représenté en traits gras) et $O_2 = \langle \alpha_0 \alpha_1 \rangle (b')$ (représenté en traits fins) sont isomorphes. En effet, il existe un isomorphisme φ de O_1 vers O_2 préservant les relations de voisinage (voir en annexe A.2.2, la définition 57 de morphisme de graphes partiellement étiquetés). Sur la figure, φ est représentée par les flèches en pointillés ; par exemple, la connexion du brin b au brin b' par une flèche en pointillés dénote $\varphi(b) = b'$. En revanche, sur la Fig. 1.26(b), les orbites $O_3 = \langle \alpha_0 \alpha_1 \rangle (b)$ (représenté en traits gras) et $O_4 = \langle \alpha_0 \alpha_1 \rangle (b')$ (représenté en traits fins) ne sont pas isomorphes. Dans l'exemple, on remarque que les brins b_1 et b_2 de O_4 ne peuvent être associés à aucun brin de O_3 .

Dans le cas des cartes généralisées, l'isomorphisme φ détermine pleinement de quelle manière les brins des orbites sont reliés lors d'une couture. Nous pouvons alors introduire l'opération de i -couture de la manière suivante.

Définition 9 (i -couture). Soient :

- $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ une carte généralisée de dimension n ;
- i un entier tel que $0 \leq i \leq n$. Notons β le mot constitué de la suite d'étiquettes α_0 à α_n privée des étiquettes α_{i-1} , α_i et α_{i+1} ;
- v et v' deux brins de V_G tels que :

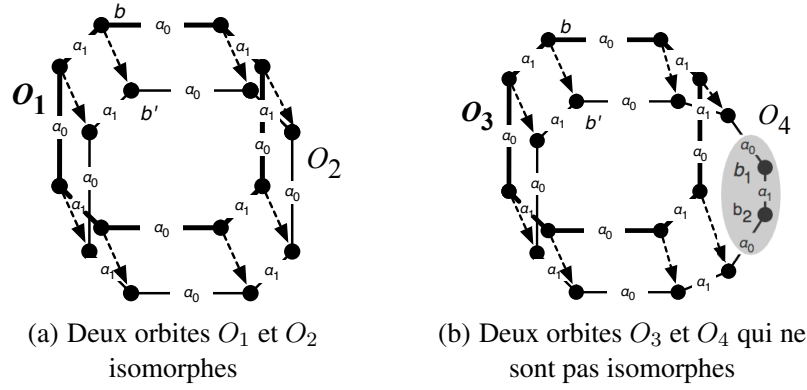


FIGURE 1.26 – Isomorphisme d'orbites

- il existe un isomorphisme φ (voir en annexe A.2.2, la définition 57 de morphisme de graphes partiellement étiquetés) de l'orbite $O = \langle \beta \rangle (v) = (V_O, E_O, s_O, t_O, l_{V_O}, l_{E_O})$ vers l'orbite $O' = \langle \beta \rangle (v') = (V_{O'}, E_{O'}, s_{O'}, t_{O'}, l_{V_{O'}}, l_{E_{O'}})$ tel que $\varphi(v) = v'$;
- pour tout brin w de V_O (respectivement pour tout brin w' de $V_{O'}$), il existe une boucle de source w et d'étiquette α_i dans E_O (il existe une boucle de source w' et d'étiquette α_i dans $E_{O'}$).

La n -G-carte $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$ sur Σ_V , résultat de la i -couture de v et v' dans G , est définie par :

- $V_H = V_G$;
- $E_H = ((E_G \setminus \{e \mid s_G(e) = t_G(e) \in V_O \text{ et } l_{E_G}(e) = \alpha_i\}) \quad (1)$
 $\setminus \{e \mid s_G(e) = t_G(e) \in V_{O'} \text{ et } l_{E_G}(e) = \alpha_i\}) \quad (2)$
 $\cup \{(w, \alpha_i, \varphi(w)) \mid w \in V_O\} \quad (3)$
 $\cup \{(\varphi(w), \alpha_i, w) \mid w \in V_{O'}\} \quad (4)$
- s_H, t_H et l_{E_H} sont respectivement les extensions canoniques de s_G, t_G et l_{E_G} sur E_H ;
- pour tout $v \in V_H, l_{V_H}(v) = \perp$.

Il est légitime de s'interroger sur le sens à donner aux orbites $\langle \beta \rangle (v)$ et $\langle \beta \rangle (v')$ mentionnés dans la définition. Rappelons que sur la Fig. 1.23, les 2-cellules F_1 et F_2 sont interprétées comme les faces le long desquelles nous souhaitons coller deux volumes V_1 et V_2 . Dans l'exemple, $\langle \beta \rangle (v)$ et $\langle \beta \rangle (v')$ sont, respectivement, les orbites $\langle \alpha_0 \alpha_1 \rangle (b_1)$ et $\langle \alpha_0 \alpha_1 \rangle (b_9)$. C'est-à-dire les 2-cellules F_1 et F_2 où l'on a « oublié » les liens α_3 , qui sont des boucles d'après la définition. Ces orbites doivent être isomorphes lorsque l'on souhaite coudre les volumes V_1 et V_2 , c'est-à-dire relier par α_3 les brins de F_1 aux brins de F_2 (voir Fig. 1.26(a)).

Commentons à présent la construction du graphe H . On remarque que les ensembles de brins V_H et V_G sont égaux. Ainsi, la couture ne modifie pas les brins de G . Les lignes (1) et (2) de la définition signifient que l'ensemble E_H des arcs de H est constitué de l'ensemble E_G des arcs de G auquel on a retiré les boucles étiquetées par α_3 incidentes aux brins des orbites $\langle \alpha_0 \alpha_1 \rangle (b_1)$ et $\langle \alpha_0 \alpha_1 \rangle (b_2)$. Enfin, les lignes (3) et (4) permettent d'ajouter dans E_H les arêtes étiquetées par α_3 qui lient b_1 et b_9 ainsi que les brins liés par l'isomorphisme φ .

1.4.4 Décousure

La découure, ou i -décousure, est l'opération inverse de la i -couture. Intuitivement, elle permet de déconnecter deux i -cellules collées le long d'une $(i-1)$ -cellule, par exemple, afin de séparer les volumes V_1 et V_2 collés le long d'une face F sur la Fig. 1.21(b). Le résultat géométrique de l'application d'une 3-décousure est illustré en Fig. 1.21(a). La découure peut être définie mathématiquement d'une manière

analogue à la couture.

Définition 10 (*i*-décousure). Soient :

- $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ une carte généralisée de dimension n ;
- i un entier tel que $0 \leq i \leq n$. Notons β le mot constitué de la suite d'étiquettes α_0 à α_n privée des étiquettes α_{i-1} , α_i et α_{i+1} ;
- v un brin de V_G ;
- l'orbite $O = \langle \beta \rangle (v) = (V_O, E_O, s_O, t_O, l_{V_O}, l_{E_O})$.

La n -G-carte $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$, résultat de la i -décousure de v dans G , est définie par :

- $V_H = V_G$;
- $E_H = (E_G$
 - $\{e \mid s_G(e) \in V_O \text{ et } l_{E_G}(e) = \alpha_i\}$ (1)
 - $\{e \mid t_G(e) \in V_O \text{ et } l_{E_G}(e) = \alpha_i\}$ (2)
 - $\cup \{ (w, \alpha_i, w) \mid w \in V_O \}$ (3)
 - $\cup \{ (w', \alpha_i, w') \mid w \in V_O \text{ et } (w, \alpha_i, w') \in E_G \}$ (4)
- s_H, t_H et l_{E_H} sont respectivement les extensions canoniques de s_G, t_G et l_{E_G} sur E_H ;
- pour tout $v \in V_H$, $l_{V_H}(v) = \perp$.

Illustrons brièvement cette définition à l'aide des Fig. 1.23 et 1.25. Prenons $i = 3$ et prenons le brin b_1 lié à b_9 par α_3 dans la Fig. 1.25. Le résultat H de la 3-décousure de la face $\langle \alpha_0 \alpha_1 \rangle (b_1)$ est donné par le graphe de la Fig. 1.23.

Comme dans le cas de la couture, on remarque que l'opération de décousure ne modifie pas les brins de G . Les lignes (1) et (2) de la définition signifient que l'ensemble E_H des arcs de H est constitué de l'ensemble E_G des arcs de G auquel on a retiré les arcs étiquetés par α_3 qui lient les brins de l'orbite $\langle \alpha_0 \alpha_1 \rangle (b_1)$ aux brins de l'orbite $\langle \alpha_0 \alpha_1 \rangle (b_9)$. Enfin, les lignes (3) et (4) permettent d'ajouter dans E_H les boucles étiquetées par α_3 incidentes aux brins ainsi déconnectés.

1.5 Opérations de haut niveau

Nous avons introduit les quatre opérations de base des cartes généralisées : l'ajout et la suppression d'un brin isolé ainsi que les opérations de couture et de décousure. Ces opérations locales permettent de construire toutes les cartes généralisées. En outre, tout mécanisme de modification d'une G-carte peut être défini comme une composition de ces opérations. Cependant, en pratique, la majorité des opérations de haut niveau doivent être définies directement afin d'être implantées efficacement.

Il est à noter que comme les opérations de base, les opérations de haut niveau doivent préserver les contraintes des cartes généralisées : non orientation, arcs incidents et cycles. Il est donc nécessaire de le vérifier pour chaque opération définie. Dans cette section, nous étudions deux de ces opérations : le *cône* [Skapin 01a, Lienhardt 04] et le *chanfreinage* [Elter 94a, Elter 94b, Ledoux 02a, Ledoux 02c].

1.5.1 Cône

Dans un modèleur géométrique, l'opération de cône fait partie des opérations qui permettent de générer un volume à partir d'une face. Ces opérations font souvent figure de pièces maîtresses dans un modèleur. En effet, le modélisateur (c'est-à-dire l'utilisateur d'un modèleur) comme l'architecte, préfèrent souvent dessiner un plan 2D de la scène 3D qu'ils souhaitent modéliser. D'une manière plus générale, l'opération de cône permet de construire une cellule topologique de dimension i à partir d'une cellule topologique de dimension $i - 1$. Notamment, il devient possible de construire une arête à partir

d'un sommet, une face à partir d'une arête, puis comme nous venons de le voir : un volume à partir d'une face. L'exemple classique de la création d'un tétraèdre à partir d'une face triangulaire est donné en Fig. 1.27. Intuitivement, il s'agit de relier les sommets A , B et C de la face F au sommet isolé D (voir Fig. 1.27(a)). Ce faisant, de nouvelles cellules topologiques sont créées (voir Fig. 1.27(b)) : les faces F_1 , F_2 et F_3 ainsi que les arêtes a , b et c .

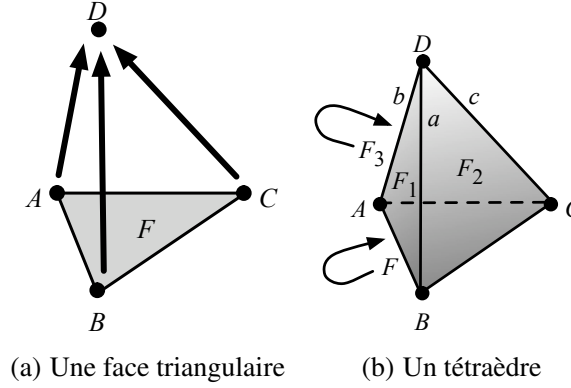


FIGURE 1.27 – Création d'un tétraèdre par application de l'opération de cône à une face triangulaire

Dans le cas des cartes généralisées, l'opération de cône peut-être vue comme un cas particulier du produit cartésien de deux G-cartes [Skapin 01a, Lienhardt 04]. Cependant, par souci de concision, nous proposons ici une définition directe de l'opération de cône.

Définition 11 (cône). Soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ une carte généralisée de dimension n et $C = (V_C, E_C, s_C, t_C, l_{V_C}, l_{E_C})$ une i -cellule – avec $i \leq n - 1$ – du bord de G , c'est-à-dire dont tout arc étiqueté par α_j – avec $i < j \leq n - 1$ – est une boucle.

Soit $V = (V_j)_{0 \leq j \leq i+1}$ une famille disjointe de nouveaux brins ($V_j \cap V_G = \emptyset$ pour tout j tel que $0 \leq j \leq i+1$ et $V_j \cap V_k = \emptyset$ pour tout j, k tels que $0 \leq j < k \leq i+1$). Pour tout j tel que $0 \leq j \leq i+1$, les brins de V_j sont des copies des brins de C tels qu'il existe une bijection $\varphi^j : V_C \rightarrow V_j$.

La n -G-carte topologique $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$, résultat du cône de C dans G est définie par :

$$V_H = V_G \cup \left(\bigcup_{j=0}^{i+1} V_j \right);$$

$$E_H = E_G \cup \begin{aligned} & - \{e \in E_C \mid l_{E_C}(e) = \alpha_{i+1}\} & (1) \\ & \cup \{(v, \alpha_{i+1}, \varphi^0(v)) \mid v \in V_C\} & (2) \\ & \cup \{(\varphi^0(v), \alpha_{i+1}, v) \mid v \in V_C\} & (3) \\ & \cup \{(\varphi^j(v), \alpha_{i-j}, \varphi^{j+1}(v)) \mid v \in V_C, 0 \leq j \leq i\} & (4) \\ & \cup \{(\varphi^{j+1}(v), \alpha_{i-j}, \varphi^j(v)) \mid v \in V_C, 0 \leq j \leq i\} & (5) \\ & \cup \{(\varphi^0(v), \alpha_k, \varphi^0(v')) \mid (v, \alpha_k, v') \in E_C\} & (6) \\ & \cup \{(\varphi^0(v'), \alpha_k, \varphi^0(v)) \mid (v, \alpha_k, v') \in E_C\} & (7) \\ & \cup \{(\varphi^j(v), \alpha_k, \varphi^j(v')) \mid (v, \alpha_k, v') \in E_C, k < i - 1 - j, 0 \leq j \leq i+1\} & (8) \\ & \cup \{(\varphi^j(v'), \alpha_k, \varphi^j(v)) \mid (v, \alpha_k, v') \in E_C, k < i - 1 - j, 0 \leq j \leq i+1\} & (9) \\ & \cup \{(\varphi^j(v), \alpha_{k+1}, \varphi^j(v')) \mid (v, \alpha_k, v') \in E_C, k > i - 1 - j, 0 \leq j \leq i+1\} & (10) \\ & \cup \{(\varphi^j(v'), \alpha_{k+1}, \varphi^j(v)) \mid (v, \alpha_k, v') \in E_C, k > i - 1 - j, 0 \leq j \leq i+1\} & (11) \\ & \cup \{(\varphi^j(v), \alpha_{i+1}, \varphi^j(v)) \mid v \in V_C, 1 \leq j \leq i+1\} & (12) \\ & \cup \{(\varphi^j(v), \alpha_k, \varphi^j(v)) \mid v \in V_C, 0 \leq j \leq i+1, i+2 < k \leq n\} & (13) \end{aligned}$$

et :

– s_H , t_H et l_{E_H} sont respectivement les extensions canoniques de s_G , t_G et l_{E_G} sur E_H ;

– pour tout $v \in V_H$, $l_{V_H}(v) = \perp$.

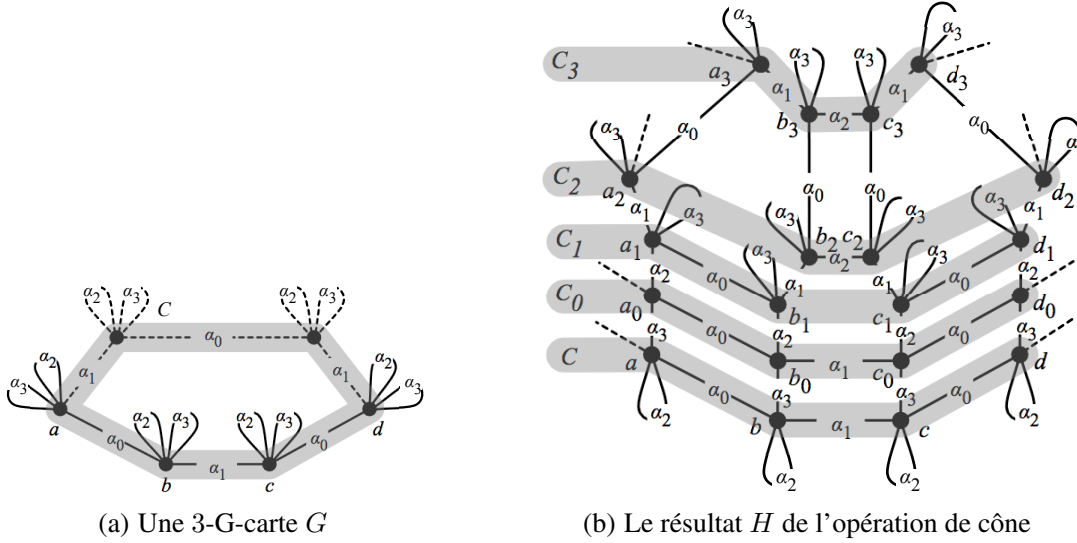


FIGURE 1.28 – Opération de cône sur une 2-cellule

L'opération de cône est illustrée en Fig. 1.28. Nous souhaitons appliquer l'opération de cône sur la 2-cellule $C = \langle \alpha_0 \alpha_1 \rangle$ (a) de la 3-G-carte G . Cette cellule – illustrée par la région grisée – est constituée des brins de G munis de tous leurs arcs incidents étiquetés par α_0 ou α_1 . Nous prenons donc $i = 2$ et $n = 3$. La 3-G-carte G modélise la face triangulaire de la Fig. 1.27(a). Le résultat H de l'opération est illustré en Fig. 1.28(b). Afin de clarifier la figure, seule une partie de H est représentée (celle qui correspond à la partie de G dessinée en traits pleins). Elle modélise la partie visible du tétraèdre de la Fig. 1.27(b), c'est-à-dire les faces F_1 et F_2 .

Dans notre cas, la première étape de l'opération consiste à créer quatre copies de C . Sur la Fig. 1.28(b), les copies sont notées C_0 , C_1 , C_2 et C_3 . On remarque que C_0 est une copie de C à l'identique (les arcs de C sont dupliqués tels quels dans C_0). En revanche, C_1 , C_2 et C_3 sont des copies de C au « renommage et suppression près des liaisons ». Par exemple, les liaisons α_0 de C sont renommées en des liaisons α_1 dans C_3 . Ainsi, les brins a et b liés par α_0 dans C sont recopiés en les brins a_3 et b_3 liés par α_1 dans C_3 . De la même manière, les liaisons α_1 de C sont renommées en des liaisons α_2 dans C_2 . Enfin, certaines liaisons de C sont supprimées. C'est le cas des liaisons α_0 de C , qui disparaissent dans C_2 . La recopie des brins de C est effectuée dans le premier point de la définition 11. Les renommages des liaisons sont effectués dans les lignes (6) à (13). Dans notre exemple, $\varphi^i(a) = a_i$, $\varphi^i(b) = b_i$, $\varphi^i(c) = c_i$ et $\varphi^i(d) = d_i$ (avec $0 \leq i \leq 3$).

Dans les lignes (1) à (5), les liaisons entre les différentes copies de C sont établies. Notamment, dans les lignes (1) et (2), les boucles α_3 de C (voir Fig. 1.28(a)) – qui nous autorisent à effectuer l'opération de cône sur C – deviennent dans H des liaisons α_3 des brins de V_C vers les brins de V_0 .

1.5.2 Chanfreinage

Dans cette section, nous présentons brièvement le chanfreinage [Elter 94a, Elter 94b, Ledoux 02a, Ledoux 02c]. Intuitivement, le chanfreinage – parfois appelé opération d'arrondi – permet d'adoucir les bords des objets. Par exemple, en appliquant le chanfreinage sur un sommet S de cube (voir Fig. 1.29(a)), on obtient le volume illustré en Fig. 1.29(b). Dans ce nouvel objet, on remarque que le sommet chanfreiné a été remplacé par une petite face F . Le fait d'appliquer plusieurs chanfreinages successifs sur un

volume permet ainsi d'adoucir ses bords.

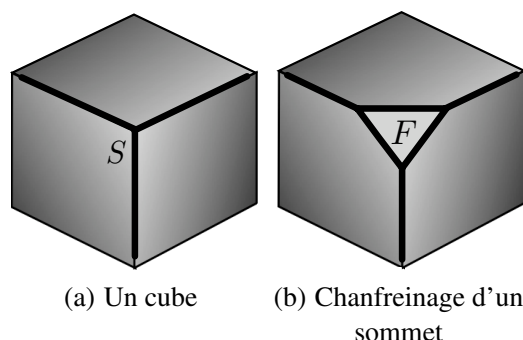
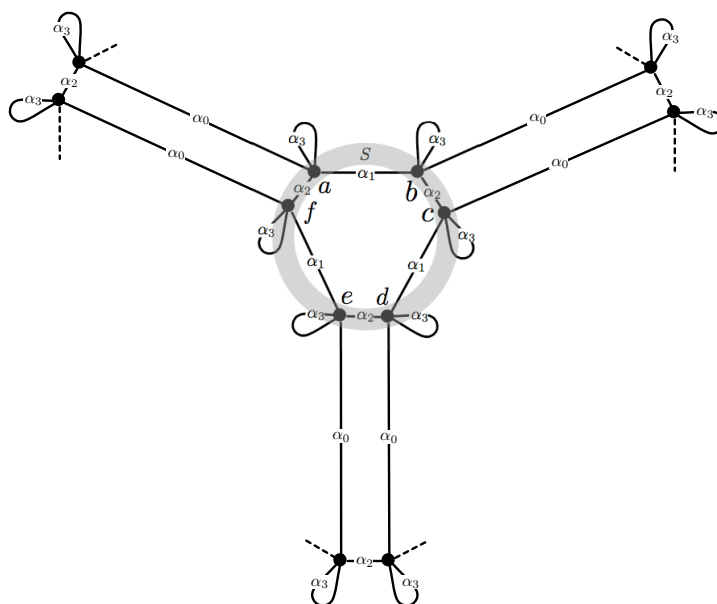
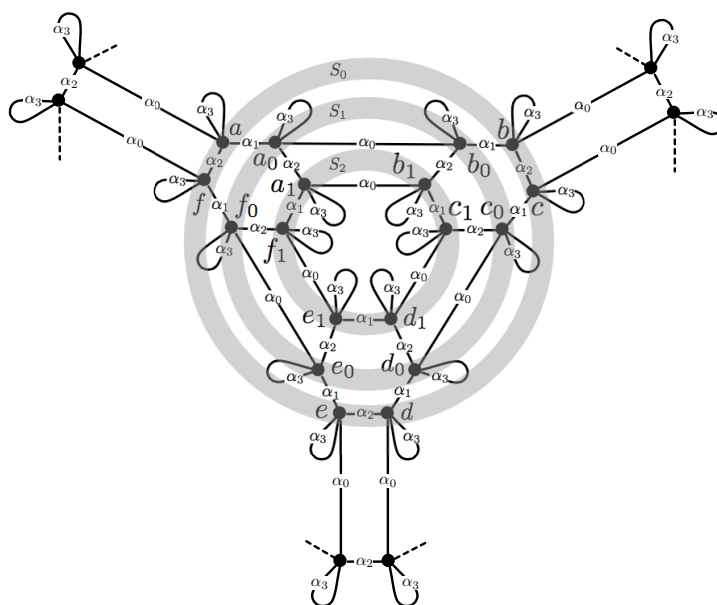


FIGURE 1.29 – Chanfreinage d'un sommet de cube

En Fig. 1.30(a), nous proposons un extrait de 3-G-carte qui représente un morceau de cube. Intuitivement, le sommet S que l'on souhaite chanfreiner est représenté par l'orbite $\langle \alpha_1 \alpha_2 \rangle$ (b) grisée. Le résultat du chanfreinage sur S est illustré en Fig. 1.30(b). Comme dans l'opération de cône, l'orbite ciblée par l'opération – dans notre cas l'orbite S – est recopiée en plusieurs exemplaires isomorphes au renommage et suppression près des liaisons. Les différentes copies de S sont illustrées par les régions grisées. Les brins de S apparaissent dans S_0 , où les liaisons α_1 ont été supprimées. Par exemple, les brins a et b connectés par α_1 dans S ne sont plus connectés dans S_0 . Dans la copie S_1 , les liaisons α_2 sont supprimées et les liaisons α_1 sont renommées en α_0 . Enfin, dans la dernière copie S_2 , les liaisons α_2 et α_1 sont renommées respectivement en α_1 et α_0 . Remarquons que la copie S_2 modélise la face F créé lors du chanfreinage (voir Fig. 1.29(b)).



(a) Extrait de la 3-G-carte d'un cube

(b) Extrait de la 3-G-carte résultat du chanfreinage d'une orbite de type $\langle \alpha_1 \alpha_2 \rangle$ FIGURE 1.30 – Opération de chanfreinage sur une orbite de type $\langle \alpha_1 \alpha_2 \rangle$

TRANSFORMATIONS DE GRAPHS ET CARTES GÉNÉRALISÉES

Sommaire

2.1	Transformations de graphes	28
2.1.1	Principe général	30
2.1.2	Principales approches	30
2.2	Approche par double somme amalgamée	32
2.2.1	Transformation directe	32
2.2.2	Condition d'arcs pendants	35
2.3	Transformations de cartes généralisées	37
2.3.1	Règles d'ajout et de suppression d'un brin isolé dans une G-carte	38
2.3.2	Satisfaction de la contrainte d'arcs incidents	39
2.3.3	Préservation des pré-G-cartes	43
2.3.4	Satisfaction de la contrainte de cycles	44
2.4	Conclusion	47

Dans le chapitre précédent, nous avons présenté le modèle des cartes généralisées ainsi que quelques-unes des opérations associées. L'opération de cône montre plusieurs limites des approches classiques de la modélisation géométrique à base topologique. En premier lieu, l'introduction d'une nouvelle opération implique l'écriture d'une définition mathématique complexe où sont décrites l'ensemble des modifications de la structure. Dans notre cas, l'ensemble des modifications des brins et des liaisons rend l'écriture – puis la lecture – de la définition du cône difficile (voir définition 11 page 22). La transcription des opérations en une définition mathématique est cependant une étape nécessaire puisqu'elle nous permet de prouver la cohérence des objets produits. En outre, il est nécessaire de produire une telle preuve pour chaque opération nouvellement définie.

Dans ce chapitre, nous proposons de définir les opérations des cartes généralisées comme des règles de *transformation de graphes* [Ehrig 06]. Cette nouvelle approche présente deux avantages. Tout d'abord, l'étape de définition d'une opération est simplifiée par le passage aux règles, qui présentent l'avantage de proposer une syntaxe graphique intuitive. Dans la pratique, définir une opération revient à « dessiner » les modifications qu'elle implique dans une G-carte, c'est-à-dire dans un graphe. De plus, nous proposons des critères syntaxiques communs à toutes les règles qui permettent de déterminer si les objets produits préservent ou non les contraintes d'arcs incidents et de cycles des cartes généralisées.

Le choix de la transformation de graphes se justifie également lors de l'étape d'implantation des opérations. Aujourd'hui, dans les modeleurs géométriques, chaque opération est implantée par un algorithme spécifique. Nous pouvons prendre pour exemple le modeleur Moka [Vidil 03], qui propose une implantation des 3-G-cartes et de leurs principales opérations. À terme, nous souhaitons définir un nouveau type de modeleur qui se présentera sous la forme d'un moteur d'application de règles. Ainsi, sous réserve d'une interface ergonomique qui permette à l'utilisateur d'entrer ses règles dans le modeleur, les étapes de définition puis d'implantation des opérations seront confondues. De plus, la correction partielle du modeleur sera assurée par construction, en vertu des critères syntaxiques communs à toutes les règles qui assurent la préservation des contraintes de cohérence des cartes généralisées.

Dans la section 2.1, nous proposons une brève introduction à la transformation de graphes. Nous évoquons notamment les différentes approches de la transformation de graphes. L'approche par *double somme amalgamée*, que nous adoptons dans ce manuscrit, est étudiée en section 2.2. Enfin, en section 2.3, nous montrons comment les opérations des cartes généralisées s'inscrivent dans le cadre de la transformation de graphes.

Dans ce chapitre, nous manipulons principalement des graphes topologiques. Rappelons qu'un graphe topologique de dimension n est un graphe partiellement étiqueté de la catégorie $\mathbb{G}e(\emptyset, \mathcal{A}_n)$, avec $\mathcal{A}_n = \{\alpha_0, \dots, \alpha_n\}$. Ainsi, pour tout graphe topologique $G = (V, E, s, t, l_V, l_E)$, la fonction d'étiquetage des sommets l_V est la fonction partout indéfinie : pour tout sommet v de G , $l_V(v) = \perp$.

2.1 Transformations de graphes

La transformation de graphes a été introduite en informatique à partir de la fin des années 60 [Pfaltz 69, Montanari 70, Pratt 71]. Le principe de base est la modification de graphes par des règles de transformation. Chaque application d'une règle sur un graphe mène à une transformation de ce graphe. Rigoureusement, le terme « transformation de graphes » désigne une séquence d'application de règles de transformation sur un graphe donné. Notons que dans la littérature, on distingue parfois les notions de transformation de graphes et de grammaire de graphes. Les grammaires de graphes sont proches des grammaires de Chomsky sur les chaînes, il s'agit d'un couple muni d'un ensemble de règles et d'un graphe de départ.

Les méthodes de même que les résultats issus des nombreux travaux sur la transformation de graphes ont été mis en pratique dans de nombreux champs de l'informatique tels que la théorie des langages, la modélisation de systèmes distribués (comme un serveur de calculs parallèles [Lin 04]), la programmation fonctionnelle [Andries 99] ou même la modélisation visuelle (où l'on représente graphiquement les acteurs d'un système et leurs interactions) [Bottoni 02]. Ce large spectre des domaines d'application de la transformation de graphes fait écho à la nature intuitive des modélisations à base de graphes.

En particulier, la transformation de graphes permet de modéliser facilement la dynamique des systèmes par une évolution de la structure graphique. Elle est ainsi devenue un outil de modélisation, de même qu'un paradigme de programmation, souvent utilisé dans le cas de systèmes complexes qui mettent en jeu de nombreuses entités communicantes. Une présentation détaillée des différentes approches des grammaires de graphes ainsi que des applications de la transformation de graphes est donnée dans [Rozenberg 97, Ehrig 99a, Ehrig 99b].

Les systèmes de réécriture de graphes comptent parmi les domaines dérivés de la transformation de graphes. De tels systèmes se rapprochent des systèmes de réécriture classiques et permettent de représenter, puis de transformer, des expressions représentées sous la forme de graphes. La représentation d'expressions par des graphes permet notamment de partager avantageusement plusieurs sous-expressions dans un même sous-graphe. Sur la Fig. 2.1 (issue de [Ehrig 99a], chapitre 1), un *hypergraphe*¹ représente graphiquement l'expression $e = s(0) + ((s(0) \times (0 + y)) + (s(0) \times (0 + y)))$. On remarque que les sous-termes $s(0)$ de e partagent un même sous-graphe mis en évidence par la zone grisée sur la figure.

Dans la suite, nous n'émettrons plus de distinction entre « grammaires de graphes », « réécriture de graphes » et « transformation de graphes ». Nous utiliserons « transformation de graphes » comme dénomination commune pour désigner la modification de graphes par des règles.

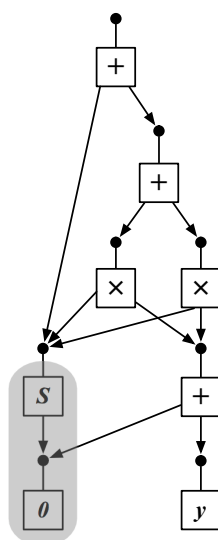


FIGURE 2.1 – Représentation par un hypergraphe de l'expression $s(0) + ((s(0) \times (0 + y)) + (s(0) \times (0 + y)))$

1. On appelle hypergraphe un graphe muni d'*hyperarcs*, c'est-à-dire muni d'arcs qui connectent un nombre quelconque de sommets. Sur la figure, les hyperarcs sont représentés par des boîtes.

2.1.1 Principe général

Le principe général de la transformation de graphes est illustré en Fig. 2.2. Le noyau d'une *règle de transformation*, ou *production*, $p = (L, R)$ est constitué de deux graphes L (représenté par un triangle) et R (représenté par un carré) appelés respectivement *partie gauche* et *partie droite* de p . Appliquer une règle p à un graphe G (à gauche sur la figure) signifie trouver un *morphisme de filtrage* m de L dans G^2 , c'est-à-dire un sous-graphe G isomorphe à L , puis le remplacer par R . À droite sur la figure, H est le graphe résultat de l'application de p au graphe G pour un filtrage m . Les principaux problèmes posés sont la suppression de L dans G puis la connexion de R dans le graphe résultat H . Les différentes solutions proposées sont à l'origine de plusieurs approches de la transformation de graphes.

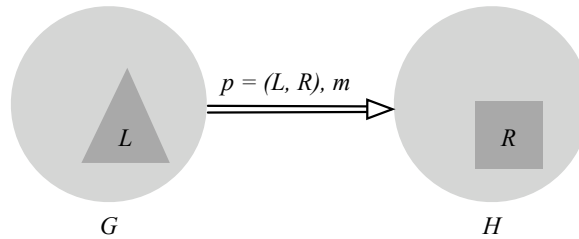


FIGURE 2.2 – Modification d'un graphe par une règle de transformation

S'il est vrai que la manière d'appliquer une règle à un graphe varie d'une approche à l'autre, il est possible d'en exhiber un algorithme général. Un tel algorithme est donné dans [Ehrig 06], nous en proposons ici une version simplifiée. Ainsi, nous décomposons la transformation d'un graphe G en un graphe H , notée $G \Rightarrow H$, en trois principales étapes. Elles sont illustrées sur la Fig. 2.3, où les flèches numérotées correspondent aux étapes.

1. Choisir une production $p : L \Rightarrow R$ composée d'une partie gauche L et d'une partie droite R et telle qu'il existe un filtrage m de L dans G .
2. Vérifier une condition d'application qui indique si la production p avec le filtrage m peut être appliquée au graphe G .
3. Supprimer dans G la partie de L qui n'est pas dans R . Cette partie est notée L' sur la figure. On appelle D le graphe ainsi obtenu.
4. Coller la partie droite R au graphe D le long d'une interface de connexion K . Cette interface est vue intuitivement comme un ensemble de sommets d'accroche qui connectent le motif L dans le graphe G . La partie de R qui n'apparaît pas dans L – on la note R' sur la figure – est ajoutée à D de manière disjointe. Le graphe H ainsi obtenu est le résultat de l'application de la production p avec le filtrage m au graphe G .

2.1.2 Principales approches

Il existe différentes manières de supprimer L de G puis de connecter R dans le graphe résultat H . Ces variantes sont à l'origine de plusieurs approches de la transformation de graphes. Les deux approches suivantes comptent parmi celles évoquées dans [Ehrig 06].

L'approche par remplacement de sommets étiquetés (ou « node label replacement approach »), développée par Rozenberg, Engelfriet et Janssens [Engelfriet 96, Janssens 83], est illustrée en Fig. 2.4(a). La partie gauche L d'une production $p : L \Rightarrow R$ est constituée d'un unique sommet s_1 . Lors de l'application

2. Nous notons m le morphisme de filtrage, en référence au terme anglais « match ».

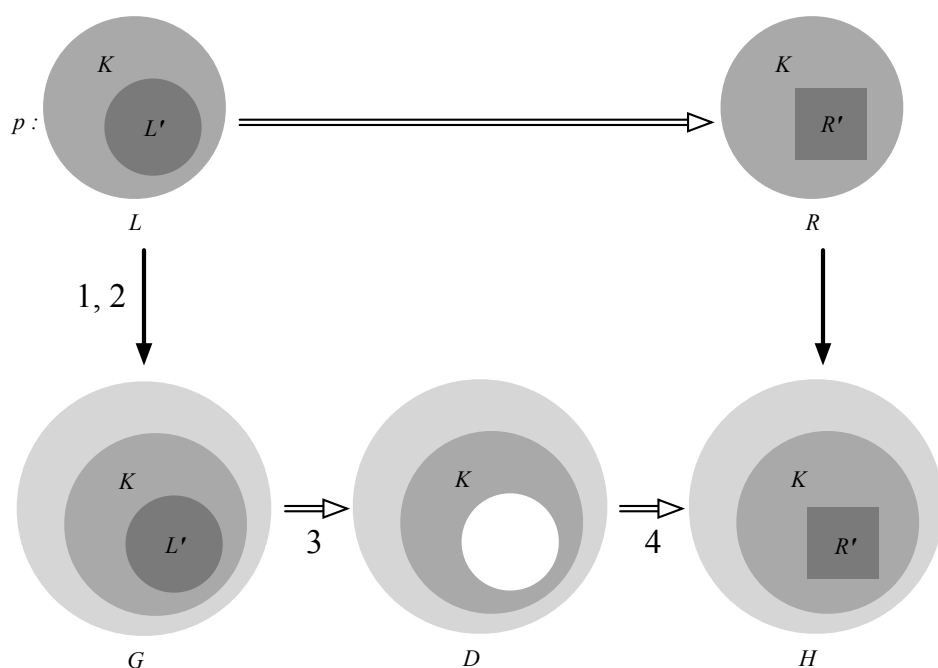


FIGURE 2.3 – Algorithme général de la transformation de graphes

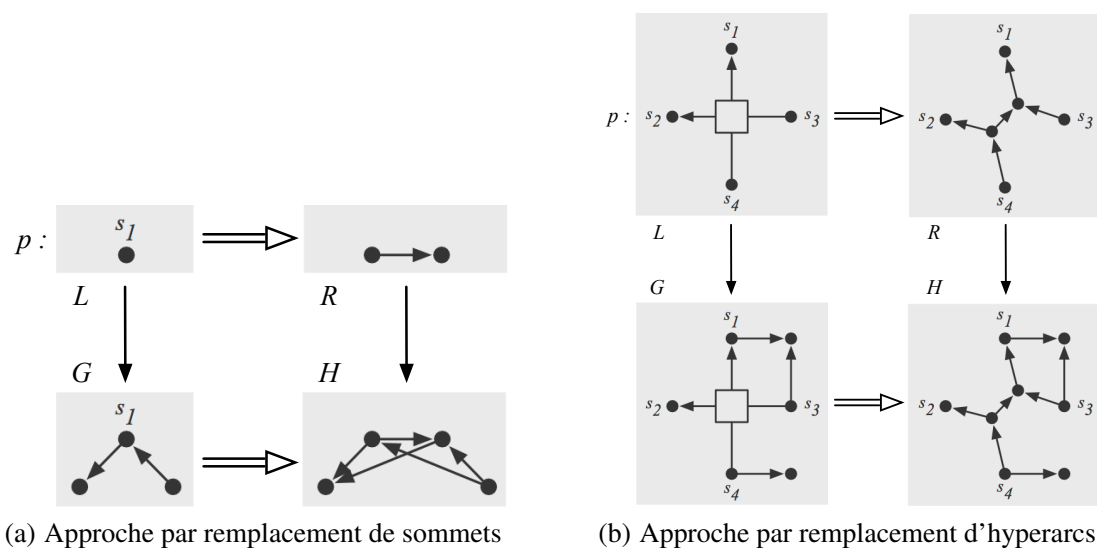


FIGURE 2.4 – Deux approches de la transformation de graphes

de la production, s_1 est remplacé par le graphe R . De plus, les arcs connectés à s_1 dans G sont connectés à tous les sommets de R dans H .

L'approche par remplacement d'hyperarcs (pour « hyperedge replacement approach », développée par Habel, Kreowski et Drewes [Drewes 96], est illustrée en Fig. 2.4(b). Ici, la partie gauche L d'une production $p : L \Rightarrow R$ contient un et un seul hyperarc muni de ses sommets sources et cibles. Lors de l'application de la production sur un graphe (ou sur un hypergraphe) G , l'hyperarc de L est remplacé par l'hypergraphe R . Sur la figure, les sommets s_1, s_2, s_3 et s_4 incidents à l'hyperarc dans L définissent l'interface qui permet de raccorder R dans H .

2.2 Approche par double somme amalgamée

Les approches citées précédemment sont parfois qualifiées d’approches ensemblistes. D’autres approches, les approches algébriques (ou « algebraic approach »), s’appuient sur la théorie des catégories. Parmi celles-ci, l’approche par produit fibré (voir en annexe A.3.4, la définition 67) propose de transformer globalement des graphes [Bauderon 01, Chen 07]. Rappelons qu’un produit fibré généralise le produit cartésien ensembliste à une catégorie quelconque. Dans le cadre de ce manuscrit, nous adoptons l’approche par *double somme amalgamée* (voir en annexe A.3.3, la définition 66), développée principalement par Ehrig et Schneider [Ehrig 06].

Dans notre cadre, les approches algébriques nous ont semblé les plus appropriées. En effet, généralement, les opérations topologiques sont utilisées afin de modifier localement une structure. Pour cette raison, nous choisissons l’approche par double somme amalgamée, dans laquelle une interface est utilisée afin de préciser où se situent les modifications à apporter aux graphes. L’approche par double somme amalgamée est donc bien adaptée pour décrire des transformations où l’on remplace localement un morceau de graphe par un autre, ce qui est le principe général des opérations topologiques.

Dans cette section, nous nous appuyons sur le premier chapitre de [Ehrig 06] afin de présenter la transformation de graphes par double somme amalgamée. Bien que nous choisissons d’illustrer cette présentation à l’aide d’exemples basées sur les G-cartes, notre discours reste général et ne tire pas parti des spécificités des G-cartes. Le positionnement de la transformation de graphes – et donc de l’approche par double somme amalgamée – par rapport aux cartes généralisées est étudié dans la section 2.3.

2.2.1 Transformation directe

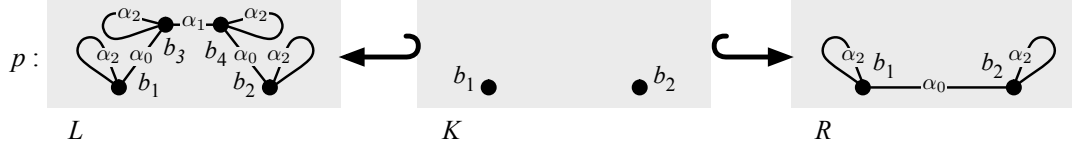
Dans l’approche par *double somme amalgamée* (ou « double-pushout »), on appelle *transformation directe* (ou encore *étape de transformation directe*) l’application d’une règle de transformation à un graphe. Dans cette section, nous commençons par décrire la transformation directe du point de vue opérationnel. En fin de section, nous donnons sa définition en termes de sommes amalgamées de la théorie des catégories.

Une production $p : L \leftarrow K \hookrightarrow R$ est la donnée de trois graphes L , K et R . La partie gauche L représente la précondition de la règle, tandis que la partie droite R décrit la postcondition. Le graphe K est l’interface, il est inclus dans L et R . Dans la pratique, K est filtré lors de l’application de p mais n’est pas modifié. En revanche, les structures $L \setminus K$ et $R \setminus K$ sont respectivement supprimées et ajoutées lors de l’application de p . Ainsi, une règle de transformation est définie de la manière suivant.

Définition 12 (production). Soient Σ_V un ensemble d’étiquettes de sommets et Σ_E un ensemble d’étiquettes d’arcs. Une règle de transformation – ou production – $p : L \leftarrow K \hookrightarrow R$ dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\Sigma_V, \Sigma_E)$ est composée de trois graphes partiellement étiquetés L , K et R de la catégorie $\mathbb{G}e(\Sigma_V, \Sigma_E)$, appelés respectivement la partie gauche, l’interface et la partie droite de p , et de deux morphismes d’inclusion de graphes partiellement étiquetés $K \hookrightarrow L$ et $K \hookrightarrow R$ (voir en annexe A.2.2, les définitions 59 et 57).

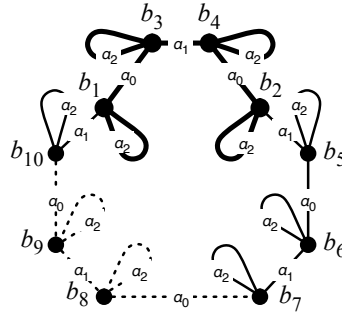
Notons que dans la littérature, on considère généralement des productions basées sur des morphismes injectifs $l : K \rightarrow L$ et $r : K \rightarrow R$ qui ne sont pas nécessairement des inclusions. Cependant, des morphismes d’inclusions sont suffisants et nous permettent de simplifier les notations. Un exemple de production p est donné en Fig. 2.5 : elle permet de remplacer deux 1-cellules connectées par α_1 , par une unique 1-cellule dans une 2-G-carte. Ici et dans la suite du manuscrit, les inclusions de K dans L et R sont explicitées par le nommage des sommets (voir en annexe A.2.2, la définition 59 des morphismes de graphes partiellement étiquetés ainsi que l’exemple associé). Intuitivement, p remplace deux arêtes

géométriques du bord de l'objet – modélisées par les 1-cellules $\langle \alpha_0 \alpha_2 \rangle (b_1)$ et $\langle \alpha_0 \alpha_2 \rangle (b_2)$ de L (voir définition 4 page 12) – connectées par un sommet géométrique commun (modélisé par la 0-cellule $\langle \alpha_1 \alpha_2 \rangle (b_3)$ de L) par une unique arête géométrique (modélisée par la 1-cellule $\langle \alpha_0 \alpha_2 \rangle (b_1)$ de R). La production p peut être utilisée, par exemple, pour transformer un pentagone en carré.

FIGURE 2.5 – Une production p

Filtrage de L dans G

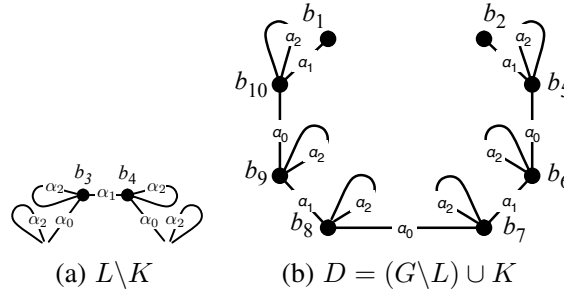
Afin d'appliquer la production p à une 2-G-carte G , il est nécessaire de filtrer le motif L dans G , c'est-à-dire trouver un sous-graphe isomorphe à L dans G . Pour cela, on construit un morphisme de filtrage $m : L \rightarrow G$. Le graphe G de la Fig. 2.6 est une 2-G-carte qui modélise la topologie d'un pentagone. On remarque qu'il existe plusieurs sous-graphes de G isomorphes à L . L'un d'entre eux est représenté en gras sur la figure. La production p peut donc être appliquée à G avec le filtrage m défini par l'inclusion dans notre exemple.

FIGURE 2.6 – Une 2-G-carte G qui modélise un pentagone

Pour choisir un autre motif, par exemple celui en pointillés, on construit le morphisme de filtrage m' qui associe les sommets b_1, b_2, b_3 et b_4 de L respectivement aux sommets b_{10}, b_9, b_8 et b_7 de G . Afin d'alléger les notations utilisées dans les exemples, nous choisissons le plus souvent le morphisme d'inclusion comme morphisme de filtrage.

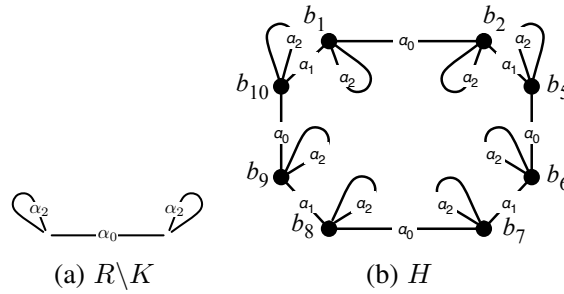
Suppression de $L \setminus K$ dans G

Lorsque la production p est appliquée à la 2-G-carte G via le morphisme de filtrage m , la première étape de la transformation directe consiste à supprimer $L \setminus K$ (voir Fig. 2.7(a)) de G . On remarque que la partie supprimée n'est pas toujours un graphe. En effet, dans notre exemple, $L \setminus K$ est muni d'arêtes pendantes étiquetées par α_0 et α_2 , car b_1 et b_2 ne sont pas supprimés. En revanche, la structure restante $D = (G \setminus L) \cup K$ (voir Fig. 2.7(b)) doit être un graphe, c'est-à-dire qu'aucune arête ne doit pendre. Pour nous assurer que D est un graphe, le morphisme de filtrage m doit satisfaire la *condition d'arcs pendants*. Afin d'alléger l'introduction de l'approche par double somme amalgamée, nous reviendrons sur cette condition dans un second temps en section 2.2.2.

FIGURE 2.7 – Suppression de $L \setminus K$ dans G

Collage de $R \setminus K$ dans D

La dernière étape de la transformation directe consiste à coller la structure $R \setminus K$ (voir Fig. 2.8(a)) dans D . À l'issue de cette étape, on obtient le graphe H résultat de la transformation directe (voir Fig. 2.8(b)). L'interface K détermine les points d'accroche de $R \setminus K$ à D . Remarquons que cette interface K possède une image dans G par la restriction du morphisme de filtrage m en raison de l'inclusion de K dans L . De plus, cette image n'est pas supprimée lors la première étape de la transformation directe. En effet, les sommets numérotés b_1 et b_2 subsistent dans D (voir Fig. 2.7(b)). Ce sont ces sommets de D issus de l'interface qui permettent de coller $R \setminus K$ à D . Dans R (voir Fig. 2.5), l'arête α_0 connecte les sommets b_1 et b_2 . Ainsi, une arête α_0 connecte également les sommets b_1 et b_2 dans H . Remarquons enfin que H est une 2-G-carte qui modélise la topologie d'un carré. Ainsi, la transformation directe de G par la production p nous a permis de transformer un pentagone en un carré.

FIGURE 2.8 – Collage de $R \setminus K$ dans G

Formalisation de la transformation directe

Nous venons de voir de manière constructive quelles sont les deux étapes d'une transformation directe qui suivent la définition du morphisme de filtrage ; à savoir la suppression de $L \setminus K$ dans G et le collage de $R \setminus K$ dans D . Dans la littérature, ces deux étapes sont introduites de manière formelle grâce aux sommes amalgamées (voir en annexe A.3, la définition 66). Ainsi, on remarque que la 2-G-carte G que l'on souhaite transformer peut être vue comme le résultat du collage de la partie gauche L au graphe D le long de l'interface K , c'est-à-dire par la somme amalgamée $G = L +_K D$ (voir le diagramme (1) de la Fig. 2.9). Remarquons que cette somme amalgamée permet de construire le graphe G à partir des graphes L et D ainsi que des morphismes de K dans L et D . Or, lors de la première étape de la transformation directe, nous cherchons à construire le graphe D . Ce graphe est obtenu par le complémentaire de la somme amalgamée $G = L +_K D$. Les conditions d'existence du complémentaire des sommes amalgamées dans la catégorie des graphes partiellement étiquetés est introduite dans le lemme 4 de [Habel 02].

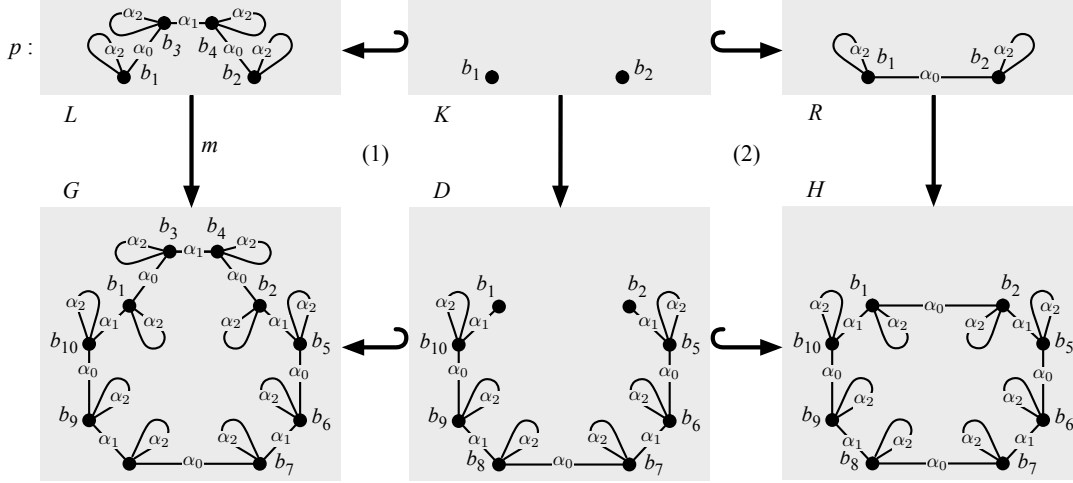
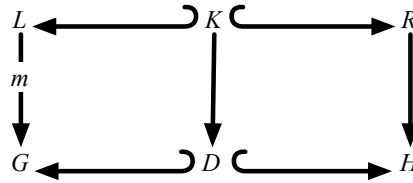


FIGURE 2.9 – Une transformation directe dans l’approche par double somme amalgamée

Dans la seconde étape, le graphe résultat H est obtenu par le collage des graphes R et D le long de K , c’est-à-dire par la somme amalgamée $H = R +_K D$ (voir le diagramme 2 de la Fig. 2.9). La transformation directe de G par la production p via le morphisme de filtrage m est alors la donnée de la double somme amalgamée de la Fig. 2.9. Enfin, une *transformation de graphes* est la donnée d’une séquence de zéro ou plusieurs transformations directes. La notion de transformation directe est définie de la manière suivante.

Définition 13 (transformation directe). Soient Σ_V un ensemble d’étiquettes de sommets et Σ_E un ensemble d’étiquettes d’arcs. Soient $p = L \leftrightarrow K \hookrightarrow R$ une production dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\Sigma_V, \Sigma_E)$, G un graphe partiellement étiqueté de $\mathbb{G}e(\Sigma_V, \Sigma_E)$ et $m: L \rightarrow G$ un morphisme de graphes partiellement étiquetés injectif appelé morphisme de filtrage.

Une transformation directe $G \xrightarrow{p,m} H$ de G en un graphe H de la catégorie $\mathbb{G}e(\Sigma_V, \Sigma_E)$ est la donnée de la double somme amalgamée suivante dans la catégorie $\mathbb{G}e(\Sigma_V, \Sigma_E)$:



Si une telle double somme amalgamée existe, on dit que p transforme G en H .

2.2.2 Condition d’arcs pendants

Nous venons de voir qu’afin d’appliquer une production $p: L \leftrightarrow K \hookrightarrow R$ sur un graphe G avec un morphisme de filtrage $m: L \rightarrow G$, il est nécessaire de construire le complémentaire de la somme amalgamée, c’est-à-dire le graphe D tel que le collage $L +_K D$ soit égal à G (voir (1) sur la Fig. 2.9). Dans un second temps, nous construisons le collage $H = R +_K D$, où H est le résultat de l’étape de transformation $G \xrightarrow{p,m} H$. Une condition de collage, que nous appelons la *condition d’arcs pendants* doit être satisfaite. Cette condition nous assure que la construction du graphe D par le complémentaire de la somme amalgamée $L +_K D = G$ est possible (voir le lemme 4 de [Habel 02]).

Définition 14 (condition d'arcs pendants). Soient Σ_V un ensemble d'étiquettes de sommets et Σ_E un ensemble d'étiquettes d'arcs. Soient $p : L \hookrightarrow K \hookrightarrow R$ une production dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\Sigma_V, \Sigma_E)$, G un graphe partiellement étiqueté de $\mathbb{G}e(\Sigma_V, \Sigma_E)$ et $m : L \rightarrow G$ un morphisme de filtrage. Le morphisme m satisfait la condition d'arcs pendants si aucun arc ne connecte un sommet non filtré de $G \setminus m(L)$ à un sommet supprimé de $m(L \setminus K)$.

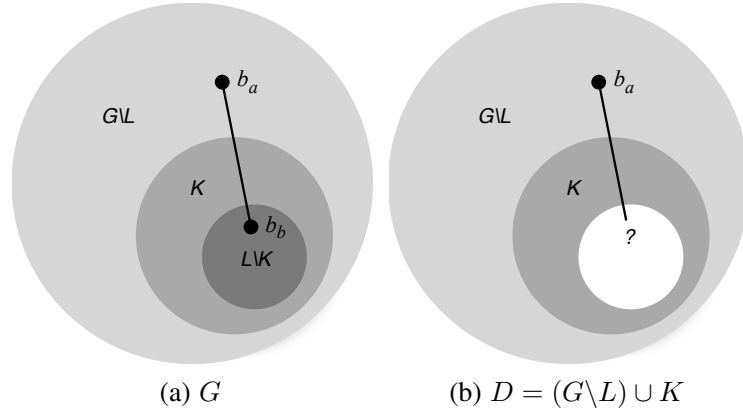
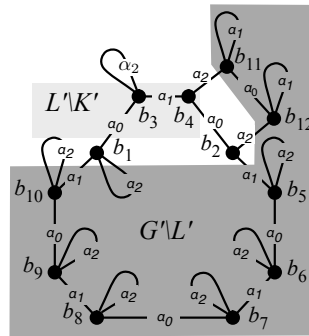
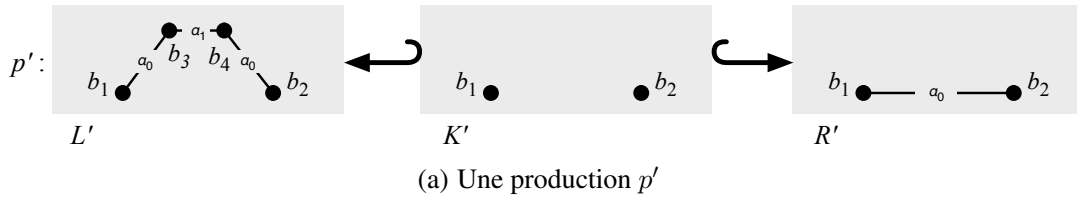


FIGURE 2.10 – Un morphisme d'inclusion m qui ne satisfait pas la condition d'arcs pendants

La condition d'arcs pendants est illustrée en Fig. 2.10. Sur la figure ainsi que dans nos commentaires, nous prenons comme morphisme de filtrage, l'inclusion. Comme dans la section précédente, les structures $m(L)$, $m(K)$ ainsi que $m(L \setminus K)$ sont donc notées respectivement L , K et $L \setminus K$. Intuitivement, le morphisme de filtrage ne satisfait pas la condition d'arcs pendants lorsqu'un sommet non filtré de $G \setminus m(L)$ est connecté à un sommet supprimé de $m(L \setminus K)$ (sur la Fig. 2.10(a), une telle connexion existe entre les sommets b_a et b_b). Dans ce cas, la structure $D = (G \setminus L) \cup K$ n'est plus un graphe car il existe un arc pendent (voir Fig. 2.10(b)).



(b) Une 2-G-carte G'

FIGURE 2.11 – Exemple dans lequel la condition d'arcs pendants n'est pas respectée

Illustrons à présent la condition d'arcs pendants sur un exemple. Nous souhaitons appliquer la production $p' = L' \leftarrow K' \hookrightarrow R'$ de la Fig. 2.11(a) à la 2-G-carte G' de la Fig. 2.11(b). La production p' est analogue à la production p de la Fig. 2.5 page 33 à ceci près que l'on n'impose pas que les brins b_1 , b_2 , b_3 et b_4 possèdent une boucle incidente étiquetée par α_2 . De même, la 2-G-carte G' est proche de la 2-G-carte G de la Fig. 2.6 page 33, mais cette fois-ci, deux nouveaux brins b_{11} et b_{12} sont respectivement connectés par α_2 aux brins b_4 et b_2 .

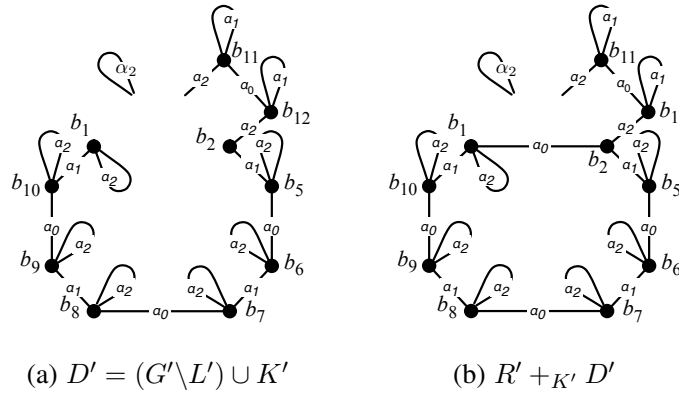


FIGURE 2.12 – La condition d'arcs pendants n'est pas respectée

Sur la Fig. 2.11(b), la partie supprimée $L' \setminus K'$ et la partie conservée $G' \setminus L'$ sont représentées, respectivement, par les zones gris clair et gris foncé. Sur cette même figure, on remarque que les sommets b_{11} et b_{12} de $G' \setminus L'$ sont respectivement connectés aux sommets b_4 et b_2 de $L' \setminus K'$. Par conséquent, la structure $D' = (G' \setminus L') \cup K'$ illustrée en Fig. 2.12(a) n'est pas un graphe. Cette violation de la condition d'arcs pendants entraîne que la structure D' construite par le complémentaire de la somme amalgamée $L' +_{K'} D' = G'$ n'est pas un graphe. En effet, l'arête α_2 incidente au sommet b_{11} reste pendante car le sommet b_4 n'appartient pas à l'interface K' le long de laquelle est effectué le collage. De même, la boucle α_2 incidente à b_3 devient pendante, car b_3 n'appartient pas non plus à l'interface K' . Poursuivre la transformation directe en effectuant le collage $R' +_{K'} D'$ aurait donc comme résultat la structure de la Fig. 2.12(b), où ces mêmes arêtes restent pendants.

Dans la section suivante, nous verrons que lorsque l'on souhaite transformer une G-carte G à l'aide d'une règle de transformation, la condition d'arcs pendants est directement liée à la condition d'arcs incidents (voir définition 1 page 10) des G-cartes. Pour rappel, dans le cas d'une n -G-carte, cette contrainte nous assure que tout sommet est la source d'exactly $n + 1$ arcs incidents étiquetés par $\alpha_0, \dots, \alpha_n$. Intuitivement, dans l'exemple de la Fig. 2.11(a), la condition d'arcs pendants nous permet de vérifier que les sommets supprimés – ici b_3 et b_4 – le sont avec tous leurs arcs incidents.

2.3 Transformations de cartes généralisées

Dans la section 1.4, nous avons présenté les quatre opérations de base des cartes généralisées : l'ajout et la suppression d'un brin isolé ainsi que la couture et la décousure de deux brins. En section 1.5, nous avons également exposé deux opérations de plus haut niveau : le cône et le chanfreinage. Ces opérations permettent de modifier la structure d'une G-carte, et donc d'un graphe. À ce titre, elles peuvent être interprétées comme des transformations de graphes, et donc *a priori* être traduites en les productions idoines.

Dans cette section, nous étudions dans un premier temps la traduction des opérations d'ajout et de suppression d'un brin isolé en règles de transformation de graphes. Dans un second temps, nous propo-

sons des critères syntaxiques sur les règles qui nous assurent que le résultat d'une étape de transformation soit une G-carte. En particulier, nos critères syntaxiques établissent un lien direct avec la contrainte d'arcs incidents (voir définition 1 page 10) et la contrainte de cycles des cartes généralisées (voir définition 5 page 14). En fin de section, nous examinons le cas des autres opérations des cartes généralisées. En particulier, nous montrons pourquoi les opérations telles que la couture nécessitent un traitement plus riche que la transformation de graphes classique présentée jusqu'ici.

2.3.1 Règles d'ajout et de suppression d'un brin isolé dans une G-carte

Ajout d'un brin isolé par transformation de graphes

Pour mémoire, l'opération d'ajout d'un brin isolé dans une n -G-carte G (voir définition 7 page 16) permet d'ajouter un brin b qui n'est connecté à aucun autre brin. Ainsi, pour tout entier i tel que $0 \leq i \leq n$, l'arête α_i incidente à b est une boucle. Dans la suite, afin de faciliter la lecture des différents exemples, nous développons les opérations pour les 3-G-cartes ($n = 3$).

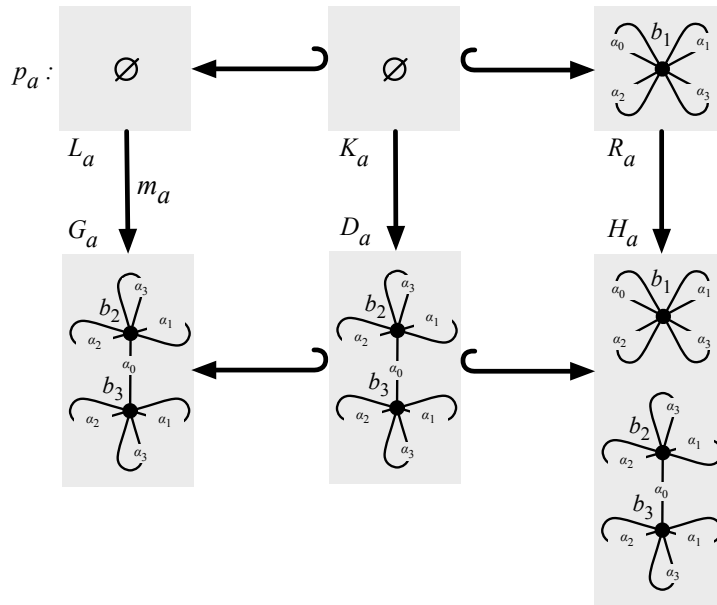


FIGURE 2.13 – Ajout d'un brin isolé dans une 3-G-carte par règle de transformation

L'opération d'ajout d'un brin isolé dans une 3-G-carte peut être traduite par la production $p_a : L_a \leftarrow K_a \hookrightarrow R_a$ de la Fig. 2.13, indexée par « a » pour « ajout ». Sur la figure, nous appliquons p_a à une 3-G-carte G_a qui modélise un objet réduit à une arête géométrique (en bas à gauche sur la figure). On remarque que la partie gauche L_a de la production p_a contient le graphe vide, c'est-à-dire le graphe dont les ensembles de sommets et d'arêtes sont vides. Par convention, nous représentons le graphe vide par le symbole \emptyset . Le graphe interface K_a – qui est inclus dans l'intersection de L_a et R_a – est également vide. Intuitivement, la partie gauche L_a vide signifie qu'aucun sous-graphe n'est filtré par p_a dans la 3-G-carte G_a . En particulier, le morphisme de filtrage m_a est trivial et peut être construit quelle que soit la 3-G-carte que l'on souhaite transformer. Ceci implique que la règle d'ajout d'un brin isolé dans une 3-G-carte peut être appliquée sur toutes les 3-G-cartes. Cette observation paraît raisonnable dans la mesure où nous souhaitons pouvoir ajouter de nouveaux brins à toutes les 3-G-cartes.

Notons que le morphisme de filtrage m_a satisfait la condition d'arcs pendants. En effet, aucune arête ne connecte un sommet de $G_a \setminus L_a = G_a$ à un sommet de $L_a \setminus K_a = \emptyset$. Ainsi, la 3-G-carte G_a est bien

le résultat du collage de $L_a = \emptyset$ et $D_a = (G_a \setminus L_a) \cup K_a = G_a$ le long de $K_a = \emptyset$. Nous avons $L_a +_{K_a} D_a = G_a$.

Rappelons que $R_a \setminus K_a$ correspond à la structure ajoutée à D_a . Plus exactement, le graphe H_a (voir Fig. 2.13) issu du collage de R_a et D_a le long de K_a , $H_a = R_a +_{K_a} D_a$, est le résultat de la transformation directe $G_a \xrightarrow{p_a, m_a} H_a$. Dans l'exemple, $R_a = R_a \setminus K_a$ contient un unique brin b_1 muni de ses boucles incidentes étiquetées par α_0 à α_3 . Le graphe H_a correspond bien à la 3-G-carte G_a à laquelle on ajoute un nouveau brin isolé.

Suppression d'un brin isolé par transformation de graphes

L'opération de suppression d'un brin isolé dans une G-carte G (voir définition 8 page 16) est l'inverse de l'opération d'ajout. Elle permet de supprimer un brin G qui n'est connecté à aucun autre brin.

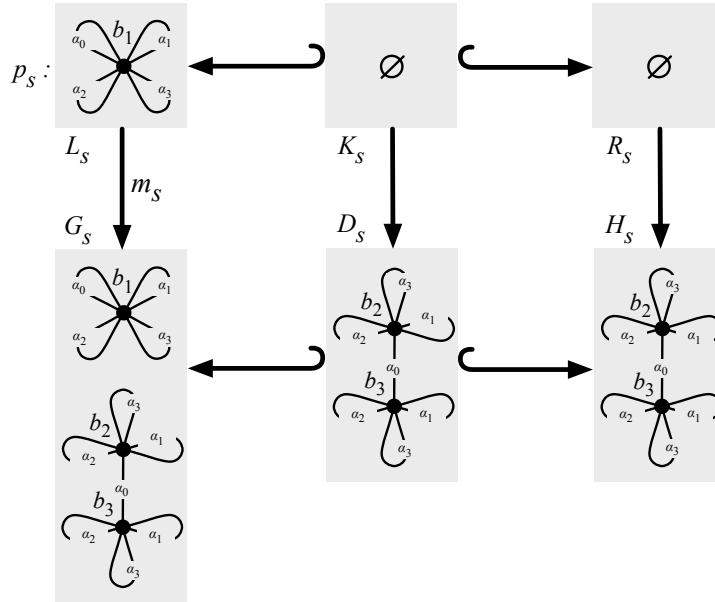


FIGURE 2.14 – Suppression d'un brin isolé dans une 3-G-carte par une règle de transformation

L'opération de suppression d'un brin isolé dans une 3-G-carte peut être définie par la production $p_s : L_s \leftarrow K_s \hookrightarrow R_s$ de la Fig. 2.14, indicée par « s » pour « suppression ». Sur la figure, nous appliquons p_s à la 3-G-carte G_s (en bas à gauche sur la figure) qui modélise une arête géométrique ainsi qu'un brin isolé. La partie gauche L_s de la production p_s contient un unique brin b_1 muni de toutes ses boucles incidentes. On remarque que l'interface K_s contient le graphe vide \emptyset .

Dans la section 2.2.1, nous avons vu que la structure $L_s \setminus K_s = L_s$ correspond à la structure supprimée dans le graphe G_s . Le graphe $D_s = (G_s \setminus L_s) \cup K_s = G_s \setminus L_s$ résulte de cette étape de suppression. La partie droite R_s de p_s est vide. Ainsi, le graphe H_s résultat du collage de R_s et D_s le long de $K_s = \emptyset$ est égal à D_s : $H_s = R_s +_{K_s} D_s = D_s$. Le graphe H_s est bien la 3-G-carte résultat de la suppression du brin isolé b_1 dans G_s .

2.3.2 Satisfaction de la contrainte d'arcs incidents

Dans cette section, nous proposons un critère syntaxique sur les règles de transformation qui nous assure que le résultat d'une transformation directe satisfait la contrainte d'arcs incidents des cartes géné-

ralisées.

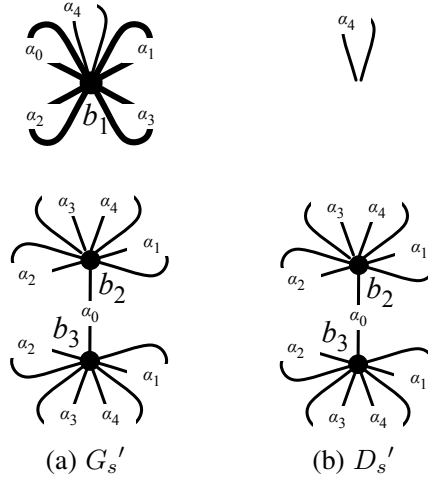


FIGURE 2.15 – En dimension 4, la condition d’arcs pendants n’est pas respectée

Remarquons que la partie gauche L_s de la règle de suppression d’un brin isolé p_s (voir Fig. 2.14) peut être filtrée par une G-carte qui n’est pas de dimension 3. En Fig. 2.15(a), nous prenons l’exemple d’une 4-G-carte G'_s . Il existe un sous-graphe isomorphe à L_s dans G'_s (il est représenté en gras sur la figure). On remarque que lorsque l’on construit le graphe $D'_s = (G'_s \setminus L_s) \cup K_s = G'_s \setminus L_s$ (voir Fig. 2.15(b)), la boucle α_4 issue du brin b_1 devient pendante. Ainsi, la condition d’arcs pendants n’est pas satisfaite lorsque le filtrage de L_s est effectué sur une 4-G-carte. Intuitivement, lorsque l’on souhaite supprimer des brins à une n -G-carte avec une règle de transformation $p : L \leftrightarrow K \hookrightarrow R$ tout en cherchant à conserver les contraintes de cohérence, les brins supprimés – c’est-à-dire les brins de $L \setminus K$ – doivent posséder leur $n + 1$ arcs incidents étiquetés par α_0 à α_n dans L . En d’autres termes, la contrainte $\text{Incede}_{L \setminus K}(\alpha_0, \dots, \alpha_n)$ doit être satisfaite.

La proposition suivante établit le lien qui existe entre la condition d’arcs pendants de la transformation de graphes et la contrainte d’arcs incidents.

Théorème 15. Soit $n \geq 0$ une dimension. Soient $p : L \leftrightarrow K \hookrightarrow R$ une production dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ telle que L et K soient des graphes non orientés (voir en annexe A.2.1, la définition 56), G un graphe partiellement étiqueté de $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ qui soit non orienté et qui satisfait la contrainte d’arcs incidents $\text{Incede}_G(\mathcal{A}_n)$ et $m : L \rightarrow G$ un morphisme de filtrage.

Le morphisme de filtrage m satisfait la condition d’arcs pendants si et seulement si $L \setminus K$ satisfait la contrainte d’arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$.

Le principal intérêt de ce théorème réside dans le fait qu’un critère syntaxique local au motif d’une règle assure que quel que soit le morphisme de filtrage m , la condition d’arcs pendants est satisfaite. La preuve de la proposition 15 s’appuie sur la remarque suivante. Intuitivement, si $L \setminus K$ – c’est-à-dire la structure supprimée – satisfait la contrainte d’arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$ alors tous les sommets supprimés sont accompagnés de tous leurs arcs incidents. Il n’y a donc pas d’arcs pendants dans D .

Démonstration. Supposons que le morphisme de filtrage m satisfait la condition d’arcs pendants. Par hypothèse, le graphe G satisfait la contrainte d’arcs incidents $\text{Incede}_G(\mathcal{A}_n)$, c’est-à-dire que tous les sommets de G sont source d’exactly $n + 1$ arcs étiquetés respectivement par α_0 à α_n (voir définition 1,

page 10). En particulier pour tout sommet v de $L \setminus K$ – c'est-à-dire pour tout sommet supprimé lors de la transformation directe $G \Rightarrow_{p,m} H$ – et pour toute étiquette α_i de \mathcal{A}_n , il existe un unique arc e de G de source v et d'étiquette α_i .

D'après la condition d'arcs pendants, e est un arc de L . Sinon, son image par filtrage $m(e)$ connecterait un sommet de $G \setminus m(L)$ à un sommet de $m(L \setminus K)$, ce qui n'est pas compatible avec la condition d'arcs pendants (voir définition 14, page 36). Or, e n'est pas un arc de K , puisque v n'est pas un sommet de K . Sinon, K ne serait pas un graphe (voir définition 12, page 32). Ainsi, e est un arc de $L \setminus K$ de source v et d'étiquette α_i pour tout sommet v de $L \setminus K$ et toute étiquette α_i de \mathcal{A}_n . La contrainte d'arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$ est satisfaite.

Réciproquement, supposons que $L \setminus K$ satisfait la contrainte d'arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$. Comme G satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$, tout arc de G dont la source est un sommet de $L \setminus K$ est un arc de $L \setminus K$. Comme L , K et G sont non orientés, par symétrie, tout arc de G dont la cible est un sommet de $L \setminus K$ est un arc de $L \setminus K$. Ainsi, le morphisme de filtrage m satisfait la condition d'arcs pendants. \square

En Fig. 2.16, nous appliquons la règle d'ajout d'un brin isolé p_a (voir Fig. 2.13) à une 4-G-carte G_a' . On remarque que l'on obtient un graphe H_a' qui ne satisfait pas la contrainte d'arcs incidents Incede des 4-G-cartes. En effet, le brin ajouté b_1 ne possède pas d'arc incident étiqueté par α_4 . Intuitivement, lorsque l'on souhaite ajouter des brins à une n -G-carte avec une règle de transformation $p : L \hookleftarrow K \hookrightarrow R$, les brins ajoutés – c'est-à-dire les brins de $R \setminus K$ – doivent posséder leur $n + 1$ arcs incidents étiquetés par $\alpha_0, \dots, \alpha_n$ dans R . Ainsi, la contrainte $\text{Incede}_{R \setminus K}(\mathcal{A}_n)$ doit être satisfaite.

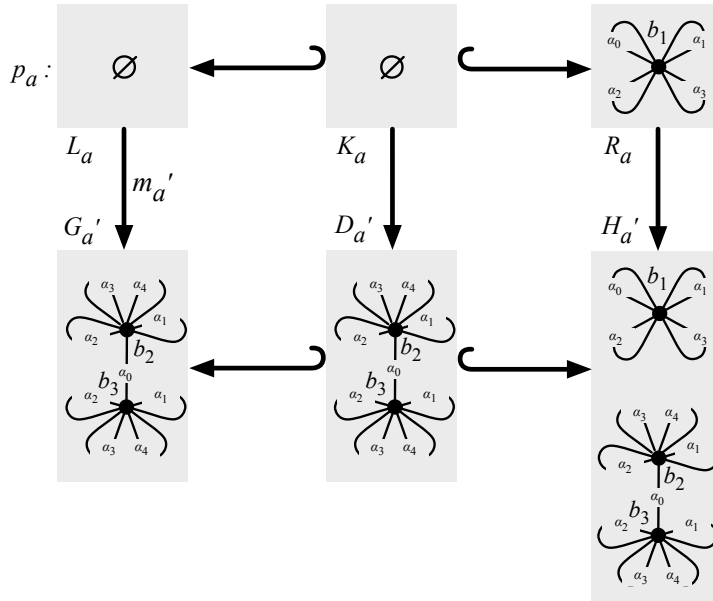


FIGURE 2.16 – Application de p_a à une 4-G-carte G'_a

Dans la proposition suivante, nous introduisons le critère syntaxique qui assure que le résultat de l'application d'une règle de transformation satisfait la contrainte d'arcs incidents.

Théorème 16. Soit $n \geq 0$ une dimension. Soit $p : L \hookleftarrow K \hookrightarrow R$ une production dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n)$, G un graphe partiellement étiqueté de $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ qui

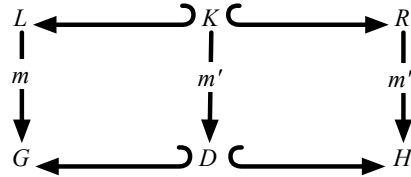
satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A})$ et $m : L \rightarrow G$ un morphisme de filtrage qui vérifie la condition d'arcs pendants.

Si la production p satisfait la contrainte d'arcs incidents $\text{Incede}_p(\mathcal{A}_n)$ suivante, alors le graphe H de $\mathbb{G}e(\emptyset, \mathcal{A}_n)$, résultat de la transformation directe $G \Rightarrow_{p,m} H$, satisfait la contrainte d'arcs incidents $\text{Incede}_H(\mathcal{A}_n)$.

$\text{Incede}_p(\mathcal{A}_n)$: les contraintes d'arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$ et $\text{Incede}_{R \setminus K}(\mathcal{A}_n)$ sont satisfaites et chaque sommet de K (c'est-à-dire chaque sommet préservé) possède les mêmes arcs incidents, c'est-à-dire pour tout sommet v de K , il existe un ensemble d'étiquettes $\mathcal{A}_n' \subset \mathcal{A}_n$ tel que les contraintes d'arcs incidents $\text{Incede}_{v,L}(\mathcal{A}_n')$ et $\text{Incede}_{v,R}(\mathcal{A}_n')$ soient toutes deux satisfaites.

Intuitivement, la satisfaction de $\text{Incede}_{R \setminus K}(\mathcal{A}_n)$ nous assure que les sommets ajoutés lors de la transformation directe sont munis de leurs $n + 1$ arcs incidents étiquetés par α_0 à α_n . De plus, comme G satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$ et comme les sommets conservés lors de la transformation – c'est-à-dire les sommets de K – conservent leurs arcs incidents, tous les sommets de H sont munis de leurs $n + 1$ arcs incidents étiquetés par α_0 à α_n .

Démonstration. Supposons que le graphe G et la production $p : L \hookleftarrow K \hookrightarrow R$ satisfont respectivement les contraintes d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$ et $\text{Incede}_p(\mathcal{A}_n)$. Par hypothèse, le morphisme de filtrage m satisfait la condition d'arcs pendants et donc la transformation directe $G \xRightarrow{p,m} H$ définie par le double diagramme ci-dessous existe :



Soit v un sommet de H :

- supposons que v ne soit l'image d'aucun sommet de R par m'' . D'après la double somme amalgamée, v est un sommet de $H \setminus m''(R)$ si et seulement si v est un sommet de $G \setminus m(L)$. De même pour les arcs, e est un arc de $H \setminus m''(R)$ de source v et d'étiquette $\alpha_i \in \mathcal{A}_n$ si et seulement si e est un arc de $G \setminus m(L)$ de source v et d'étiquette α_i . Or, G satisfait la contrainte d'arcs incidents $\text{Incede}_{v,G}(\mathcal{A}_n)$ et donc H satisfait la contrainte d'arcs incidents $\text{Incede}_{v,H}(\mathcal{A}_n)$;
- supposons que v soit un sommet ajouté lors de la transformation directe, c'est-à-dire un sommet de $m''(R \setminus K)$. Soit α_i une étiquette de \mathcal{A}_n , comme la condition d'arcs incidents $\text{Incede}_{R \setminus K}(\mathcal{A}_n)$ est satisfaite et que m'' est injectif, v possède exactement un arc incident étiqueté par α_i dans $m''(R \setminus K)$. Le double diagramme de la définition 13 nous indique que v et son unique arc incident étiqueté par α_i appartiennent également à H . De plus, comme v n'est pas un sommet de $m'(K)$ – et donc pas non plus un sommet de D ni de G – il n'existe pas d'autre arc incident à v et étiqueté par α_i dans H . De plus, H n'a pas d'autre arc incident à v . Ainsi, la contrainte d'arcs incidents $\text{Incede}_{v,H}(\mathcal{A}_n)$ est satisfaite ;
- supposons que v soit un sommet filtré et conservé lors de la transformation directe, c'est-à-dire qu'il existe un sommet u de K – et de L et R – tel que $m(u) = v$. D'après la contrainte d'arcs incidents $\text{Incede}_p(\mathcal{A}_n)$, il existe $\mathcal{A}_n' \subset \mathcal{A}_n$ tel que $\text{Incede}_{u,L}(\mathcal{A}_n')$ et $\text{Incede}_{u,R}(\mathcal{A}_n')$ soient satisfaites. Soit α_i une étiquette de \mathcal{A}_n , deux cas existent :
 - Supposons que $\alpha_i \in \mathcal{A}_n'$. Il existe un unique arc de source u et d'étiquette α_i dans L . De même, il existe un unique arc de source u et d'étiquette α_i dans R , dont l'image par m'' est un arc de H ;

- Supposons que $\alpha_i \notin \mathcal{A}_n'$. D'après la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$, il existe un unique arc e de source v et d'étiquette α_i dans G . Or, e n'est pas filtré, c'est-à-dire qu'il n'est l'image d'aucun arc de L par m . Donc e est un arc de H ;

De plus, H n'a pas d'autre arc incident à v . Ainsi, la contrainte d'arcs incidents $\text{Incede}_{v,H}(\mathcal{A}_n)$ est satisfaite.

La contrainte d'arcs incidents $\text{Incede}_H(\mathcal{A}_n)$ est donc satisfaite. \square

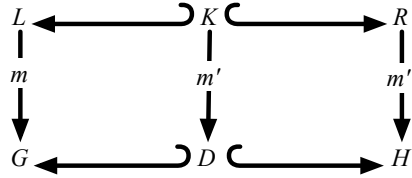
2.3.3 Préservation des pré-G-cartes

Nous venons d'introduire un critère syntaxique sur les règles de transformation qui nous assure que le graphe résultat d'une transformation satisfait la contrainte d'arcs incidents. Dans la définition 2 page 2, nous avons vu que les pré-G-cartes sont des graphes non orientés. Dans cette section, nous introduisons une condition syntaxique sur les règles qui nous assure que le graphe résultat d'une transformation est également non orienté.

Proposition 17. *Soit $n \geq 0$ une dimension. Soit $p : L \leftrightarrow K \hookrightarrow R$ une production dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n)$, G un graphe de $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ et $m : L \rightarrow G$ un morphisme de filtrage qui vérifie la condition d'arcs pendants.*

Si p est une règle non orientée – c'est-à-dire telle que L , K et R sont des graphes non orientés – et G est un graphe non orienté, alors le graphe H de la catégorie $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ résultat de la transformation directe $G \Rightarrow_{p,m} H$ est un graphe non orienté.

Démonstration. Supposons que H soit construit par le double diagramme suivant :



Soit e un arc de H de source u , de cible v et d'étiquette α_i . Montrons que H a un arc inverse f de source v , de cible u et d'étiquette α_i .

Supposons que e n'est l'image d'aucun arc de R par m'' . Donc e est un arc de G . Or G a un arc f inverse de e , et e n'est pas filtré (e n'est l'image d'aucun arc de L par m). Donc f n'est pas non plus filtré, puisque L est non orienté. Donc f est un arc de H .

Supposons que e est l'image d'un arc e' de R par m'' . Comme R est non orienté, R a un arc f' inverse de e' et l'image de f' par m'' est un arc f inverse de e .

Donc H est non orienté. \square

Nous savons à présent écrire des règles telles que leur application sur une pré-G-carte construise toujours une pré-G-carte.

Corollaire 18. *Soient G une pré-G-carte topologique de dimension n , $p : L \leftrightarrow K \hookrightarrow R$ une production dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ et $m : L \rightarrow G$ un morphisme de filtrage.*

Si p est une règle non orientée qui satisfait la contrainte d'arcs incidents (voir théorème 16), alors le graphe H résultat de la transformation directe $G \Rightarrow_{p,m} H$ existe et est une pré-G-carte de dimension n .

Démonstration. Découle directement de la satisfaction de la condition d'arcs pendants (voir théorème 15), de la préservation de la contrainte d'arcs incidents (voir théorème 16) et de la préservation de la non orientation (voir proposition 17). \square

2.3.4 Satisfaction de la contrainte de cycles

Dans cette section, nous introduisons le critère syntaxique que doit vérifier une production afin d'assurer que le résultat d'une transformation directe soit une carte généralisée. Cette contrainte inclut la satisfaction de la contrainte d'arcs incidents sur les productions de même que le critère de non orientation proposé dans la section précédente. Nous devons à présent proposer un critère syntaxique tel que le graphe résultat d'une transformation directe satisfait la contrainte de cycles (voir définition 5, page 14). Introduisons ce nouveau critère syntaxique à l'aide de trois exemples de productions.



FIGURE 2.17 – Règle de fusion de deux arêtes dans une 2-G-carte

Dans un premier temps, revenons à la production $p : L \leftarrow K \hookrightarrow R$ de la Fig. 2.17 – déjà étudiée en section 2.2.1 – qui permet de fusionner deux arêtes dans une 2-G-carte. On remarque dans un premier temps que p est non orientée et satisfait la contrainte d'arcs incidents. En effet, d'une part L , K et R sont non orientés. D'autre part, les brins ajoutés b_3 et b_4 possèdent tous leurs arcs incidents dans L . Dans le cas présent, ils possèdent tous leurs arcs incidents étiquetés par α_0 , α_1 et α_2 . De plus, les brins conservés b_1 et b_2 de K possèdent les mêmes arcs incidents – ici α_0 et α_2 – dans la partie gauche L et dans la partie droite R . Ainsi, le graphe H résultat de la transformation directe $G \Rightarrow_{p,m} H$ d'une G-carte, est une pré-G-carte.

On remarque également que le graphe H satisfait la contrainte de cycles. En effet, dans H , les brins b_1 , b_2 et b_5 à b_{10} apparaissent tous dans un cycle $\alpha_0\alpha_2\alpha_0\alpha_2$, qui est l'unique cycle sur lequel porte la contrainte de cycles dans le cas d'une 2-G-carte. Intuitivement, la préservation de la contrainte de cycles dans H est assurée par le fait que les brins b_{10} et b_5 , qui apparaissent chacun dans un cycle $\alpha_0\alpha_2\alpha_0\alpha_2$ dans L , apparaissent encore dans un tel cycle dans R . Plus clairement, les brins impliqués dans un cycle dans L doivent tous être impliqués dans un cycle analogue – c'est-à-dire étiqueté par le même mot – dans R .

Étudions à présent la production $P_{inv} : L_{inv} \leftarrow K_{inv} \hookrightarrow R_{inv}$ de la Fig. 2.18. Remarquons que p_{inv} est la production p dans laquelle nous avons inversé les parties gauche L et droite R . Elle permet donc de scinder une arête en deux dans une 2-G-carte en ajoutant une nouvelle 0-cellule. Comme p , p_{inv} est non orientée et satisfait la contrainte d'arcs incidents. Ainsi, le graphe H_{inv} résultat de la transformation directe $G_{inv} \Rightarrow_{p,m} H_{inv}$ (avec G_{inv} une 2-G-carte) existe, est non orienté, et satisfait la contrainte d'arcs incidents. Le graphe résultat H_{inv} satisfait également la contrainte de cycles des 2-G-cartes. Intuitivement, la contrainte de cycles est satisfaite car les sommets ajoutés b_3 et b_4 apparaissent tous les deux dans un cycle $\alpha_0\alpha_2\alpha_0\alpha_2$ dans R . Ainsi, les brins ajoutés par une production doivent tous être munis de leurs cycles dans la partie droite. Et comme précédemment, les brins b_1 et b_2 de l'interface se trouvent dans R dans un cycle analogue à celui filtré dans L .

Étudions enfin la production $P_{cou} : L_{cou} \leftarrow K_{cou} \hookrightarrow R_{cou}$ de la Fig. 2.19. Cette production permet de coudre par α_3 deux volumes le long de faces triangulaires dans une 3-G-carte. Ainsi, le motif L_{cou} représente deux demi-faces $\langle \alpha_0\alpha_1 \rangle (b_1)$ et $\langle \alpha_0\alpha_1 \rangle (b_7)$ qui ne possèdent pas de voisine par α_3

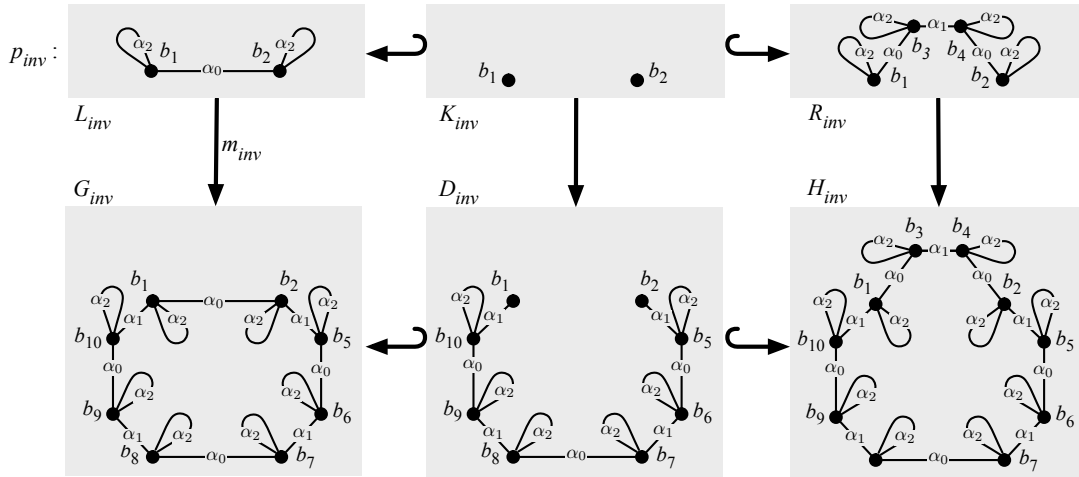
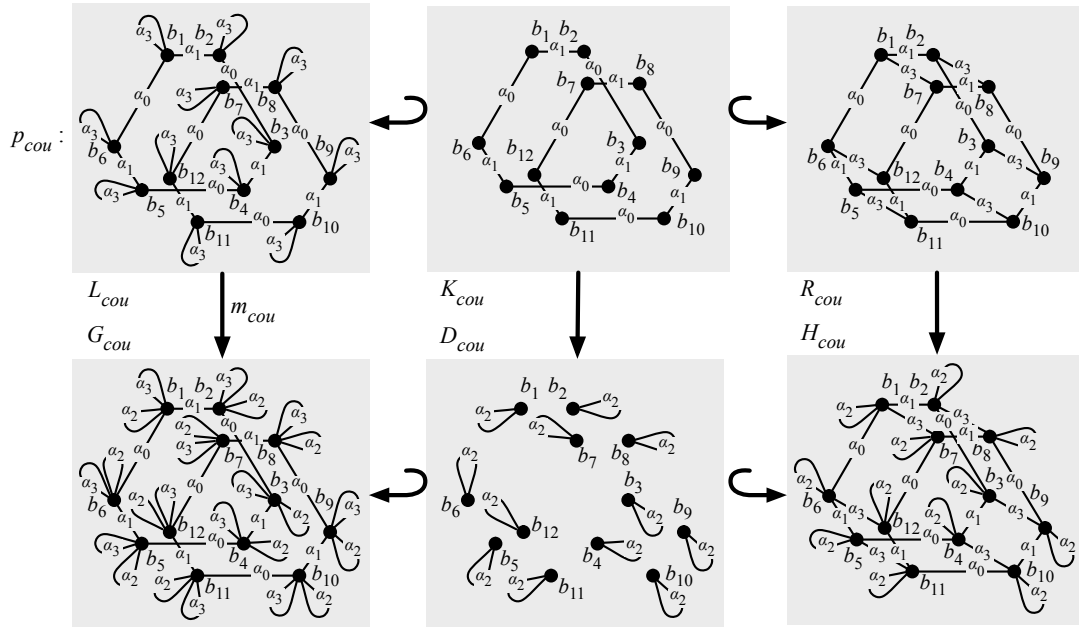


FIGURE 2.18 – Règle de scission d'une arête dans une 2-G-carte

FIGURE 2.19 – Couture par α_3 de deux faces triangulaires

(on dit que leurs brins sont α_3 -libres). Dans la partie droite R_{cou} , les deux demi-faces sont réunies en une seule 2-cellule grâce aux liaisons α_3 qui connectent les brins b_1 à b_6 respectivement aux brins b_7 à b_{12} . Afin de ne pas surcharger la figure, la 3-G-carte G_{cou} modélise deux volumes ouverts – leurs brins sont tous α_2 -libres – qui sont réduits aux deux faces à coudre.

À nouveau, la production p_{cou} est non orientée et satisfait la contrainte d'arcs incidents et donc le graphe H_{cou} résultat de la transformation directe $G_{cou} \Rightarrow_{p,m} H_{cou}$ existe, est non orienté et satisfait la contrainte d'arcs incidents. De plus, le graphe H_{cou} est une 3-G-carte car il satisfait également la contrainte de cycles des 3-G-cartes. Rappelons que dans une 3-G-carte, chaque brin doit notamment apparaître dans un cycle $\alpha_0\alpha_2\alpha_0\alpha_2$. Dans la partie gauche L_{cou} de la production p_{cou} , les liaisons α_2 ne sont pas filtrées. En revanche, les liaisons α_0 apparaissent. Ainsi, pour ne pas risquer de rompre les cycles $\alpha_0\alpha_2\alpha_0\alpha_2$, les liaisons α_0 ne sont pas modifiées par p_{cou} . Ainsi, nous imposons dans la contrainte

de cycle des règles que les liaisons impliquées dans des cycles, et dont les cycles sont incomplets dans la partie gauche, soient conservées telles quelles dans la partie droite. Notons que ce choix, qui paraît contraignant, s'est révélé approprié pour l'ensemble des règles de transformations traitées dans ce manuscrit, et donc des opérations topologiques qu'elles traduisent.

La proposition suivante regroupe les observations que nous venons d'effectuer dans nos trois exemples.

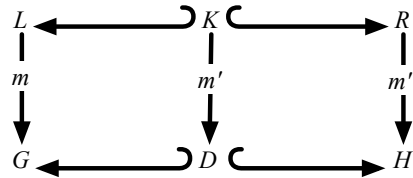
Théorème 19. Soit $n \geq 0$ une dimension. Soit $p : L \leftrightarrow K \hookrightarrow R$ une production dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ non orientée qui satisfait la contrainte d'arcs incidents $\text{Incid}_p(\mathcal{A}_n)$ et G une n -G-carte. Le graphe H résultat de la transformation directe $G \Rightarrow_{p,m} H$ est une n -G-carte si la production p satisfait la contrainte de cycles $\text{Cycle}_p(\mathcal{A}_n)$ suivante.

$\text{Cycle}_p(\mathcal{A}_n)$: pour tout couple d'entiers (i, j) qui satisfait $0 < i + 2 \leq j \leq n$,

1. soit v un sommet conservé de K . S'il existe un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v dans L alors il existe un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v dans R ;
2. soit v un sommet ajouté de $R \setminus K$. Il existe un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v dans R ;
3. soit v un sommet de L . S'il n'y a pas de cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v dans L , alors tout arc e de L de source v et d'étiquette α_i ou α_j est également un arc de R .

On remarque que les points 1, 2 et 3 de la proposition 19 sont respectivement liés aux contraintes que nous avons introduites au sujet des productions p , p_{inv} et p_{cou} .

Démonstration. D'après le corollaire 18, H existe et est une pré-G-carte. Montrons que H satisfait la contrainte de cycles $\text{Cycle}_H(\mathcal{A}_n)$. Nous supposons que H est construit par le double diagramme suivant :



Soient v un sommet de H et (i, j) un couple d'entiers qui satisfait $0 < i + 2 \leq j \leq n$:

- supposons que v soit un sommet non filtré de $G \setminus m(L)$. Comme G satisfait la contrainte de cycles, il existe un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v dans G :
 - supposons qu'aucun arc du cycle ne soit filtré (ce sont tous des arcs de $G \setminus m(L)$). Ainsi, par transformation directe, v est source du même cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ dans H ;
 - supposons que l'un des arcs du cycle soit filtré, c'est-à-dire soit un arc de $m(L)$. Comme v n'est pas un sommet filtré (v n'est pas un sommet de $m(L)$), tous les arcs du cycle ne sont pas filtrés. Nous sommes donc dans le cas 3 du théorème 19 et donc les arcs du cycle filtrés – ceux de $m(L)$ – ne sont pas modifiés. Le sommet v est donc la source du même cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ dans H ;
- supposons que v est un sommet conservé (il existe un sommet u de K tel que $v = m'(u)$). Par hypothèse, v est la source d'un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ dans G :
 - supposons que tout ce cycle ne soit pas filtré, c'est-à-dire que certains arcs du cycle n'appartiennent pas à $m(L)$. Dans ce cas, comme précédemment, le point 3 du théorème 19 nous assure que le même cycle appartient à H ;
 - supposons que tout ce cycle soit filtré. Dans ce cas, d'après le point 1 du théorème 19, il existe un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source u dans R . Par transformation directe, son image par m'' dans H est un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v ;

- supposons enfin que v soit un sommet ajouté (il existe un sommet u de $R \setminus K$ tel que $v = m''(u)$). Dans ce cas, le point 2 du théorème 19 nous assure qu'il existe un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source u dans R . Par transformation directe, ce cycle a pour image par m'' un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v dans H .

Donc pour tout sommet v de H et tout couple d'entiers (i, j) tel que $0 < i + 2 \leq j \leq n$, H possède un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ de source v . Donc H satisfait la contrainte de cycles $Cycle_H(\mathcal{A}_n)$. Par conséquent, H est une n -G-carte. □

Ainsi, ce théorème – qui reprend les critères syntaxiques définis dans le théorème 16 et la proposition 17 et introduit un nouveau critère syntaxique lié à la contrainte de cycles – nous permet d'écrire des règles de transformation dont l'application à une n -G-carte construit toujours une n -G-carte.

2.4 Conclusion

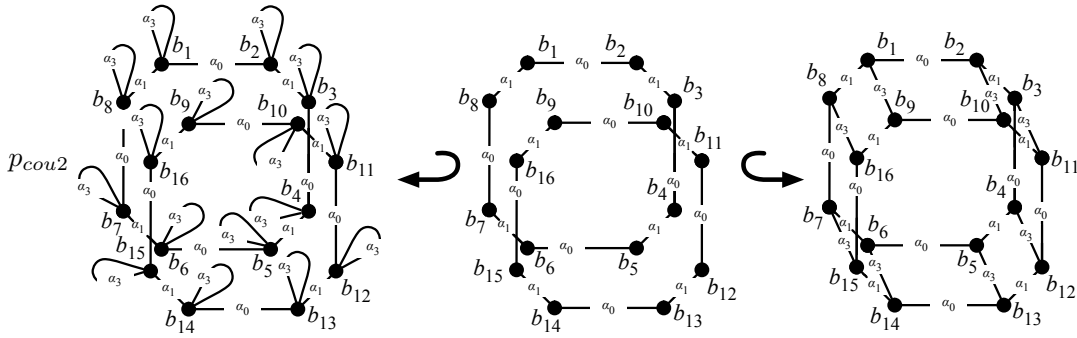


FIGURE 2.20 – Règle de couture par α_3 de deux faces carrées

Dans ce chapitre, nous avons montré comment écrire des règles qui permettent de transformer une n -G-carte en une n -G-carte. Ce faisant, nous pouvons traduire les opérations topologiques des cartes généralisées en règles de transformation de G-cartes. Cependant, nous ne pouvons encore traduire l'ensemble des opérations usuelles en une unique règle. Illustrons cette observation à l'aide des Fig. 2.19 et 2.20. Les deux productions p_{cou} et p_{cou2} permettent de coudre deux volumes le long d'une face. L'une comme l'autre représente donc une variante de l'opération de 3-couture. Dans la Fig. 2.19, la 3-couture est effectuée le long d'une orbite $\langle \alpha_0 \alpha_1 \rangle$ triangulaire tandis que dans la Fig. 2.20, elle est effectuée le long d'une orbite $\langle \alpha_0 \alpha_1 \rangle$ carrée. Intuitivement, il existe autant de variantes de la 3-couture – et donc autant de règles de 3-couture sous-jacentes – que de types d'orbites $\langle \alpha_0 \alpha_1 \rangle$, c'est-à-dire une infinité.

Dans le chapitre suivant, nous étendons nos règles afin notamment de pouvoir écrire une règle de 3-couture indépendante de la structure de l'orbite $\langle \alpha_0 \alpha_1 \rangle$ considérée. Pour cela, nous définissons de nouvelles règles paramétrées par des orbites, par exemple un triangle ou un carré.

TRANSFORMATIONS DE GRAPHS AVEC VARIABLES ET CARTES GÉNÉRALISÉES

Sommaire

3.1	Principe général de la transformation de graphes avec variables	50
3.1.1	Règles de transformation avec variables	50
3.1.2	Instanciation d'une règle de transformation avec variables	51
3.1.3	Application d'une règle de transformation avec variables	51
3.1.4	Variables graphes et clones	53
3.2	Variables orbites	57
3.2.1	Règles de transformation avec variable orbite	57
3.2.2	Instanciation des variables orbites	59
3.2.2.1	Élément neutre et saturation	60
3.2.2.2	Instanciation des graphes avec variable orbite	64
3.2.2.3	Instanciation des règles de transformation avec variable orbite	66
3.3	Renommages d'orbites	68
3.3.1	Motivations	68
3.3.2	Renommages d'arcs dans les graphes	72
3.3.3	Renommages et effacements d'arcs dans les graphes	73
3.3.4	Règles de transformation avec renommages d'orbites	75
3.3.5	Instanciation des renommages d'orbites	77
3.3.5.1	Saturation et graphe anonyme	78
3.3.5.2	Instanciation des graphes avec renommages d'orbites	79
3.3.5.3	Équivalence des graphes avec renommages et des graphes avec variable orbite	84
3.3.5.4	Instanciation des règles de transformation avec renommages	87
3.3.5.5	Chanfreinage topologique	87
3.4	Renommage d'orbites et cohérence des cartes généralisées	88
3.4.1	Contrainte d'arcs incidents	88
3.4.2	Préservation des pré-G-cartes	92
3.4.3	Condition d'arcs pendants	94
3.4.4	Contrainte de cycles	94

Dans le chapitre précédent, nous avons défini des règles qui transforment les cartes généralisées. Pour cela, nous avons défini des critères syntaxiques sur les règles qui assurent la préservation des contraintes de cohérence des cartes généralisées. Nous savons par exemple écrire une règle qui, dans une 3-G-carte, colle deux volumes le long d'une face triangulaire (voir la production p_{cou} de la Fig. 2.19 page 45). Nous savons également écrire une règle analogue qui colle deux volumes le long d'une face carrée (voir la production p_{cou2} de la Fig. 2.20 page 47). En revanche, notre formalisme ne nous permet pas d'écrire une règle qui effectue la couture de deux volumes le long d'une face quelconque, ce qu'exprime pourtant la définition classique de l'opération de couture (voir définition 9, page 19).

Dans ce chapitre, afin de lever cette limite de notre approche, nous paramétrons nos règles par une orbite. Nous pouvons ainsi définir la couture de manière générique en utilisant une variable pour dénoter les faces à coller. En substituant la variable par une face triangulaire, nous obtenons alors la règle p_{cou} précédemment définie. Nous appelons ces nouvelles règles les *règles de transformation avec variable orbite*. Pour cela, nous nous appuyons sur les *règles de transformation avec variables* présentées dans [Hoffmann 05]. Celles-ci s'appuient sur la notion de grammaire à deux niveaux [Cleaveland 77]. Intuitivement, il s'agit dans un premier temps d'écrire une règle munie de variables, puis dans un second temps d'instancier cette règle en une règle de transformation classique, c'est-à-dire sans variable. Dans notre cas, les variables dénotent des orbites. Ainsi, dans le cas de la couture de deux volumes, une règle munie d'une variable instanciée par une orbite triangulaire de type $\langle \alpha_0 \alpha_1 \rangle$ nous permet de générer la règle de couture de deux volumes le long de faces triangulaires. À l'inverse, la même règle munie d'une variable instanciée à présent par une orbite carrée du même type $\langle \alpha_0 \alpha_1 \rangle$ nous permet de générer la règle de couture de deux volumes le long de faces carrées.

Dans la section 3.1, nous présentons le principe général de la transformation de graphes avec variables introduit par B. Hoffmann. Nous présentons notamment deux types de variables – les *variables graphes* et les *variables clones* – dont nous nous inspirons pour définir nos *variables orbites*. Dans la section 3.2, nous présentons nos variables orbites et les illustrons dans le cadre de l'opération de couture de deux volumes. Dans la section 3.3, nous présentons nos règles de transformation avec renommages d'orbites. Dans celles-ci, des opérations de renommage nous permettent de dupliquer des orbites tout en supprimant et/ou en renommant leurs liaisons. De cette manière, nous pouvons encoder les opérations de haut niveau telles que l'opération de cône ou de chanfreinage, où la duplication d'orbites avec renommages joue un rôle essentiel. Dans la section 3.4, nous introduisons également les critères syntaxiques qui nous assurent que les règles avec variable orbite et avec renommages d'orbites préservent les contraintes de cohérence des G-cartes : la non orientation, la contrainte d'arcs incidents et la contrainte de cycles.

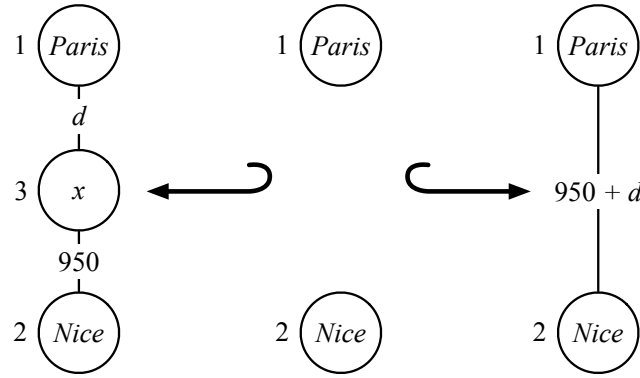
Notons qu'une version préliminaire des travaux que nous exposons dans ce chapitre a fait l'objet de deux publications [Poudret 07b, Poudret 08a].

3.1 Principe général de la transformation de graphes avec variables

Dans cette section introductive, nous présentons succinctement les principes généraux de la transformation de graphes avec variables introduite par B. Hoffmann dans [Hoffmann 05]. En particulier, nous montrons comment les variables interviennent dans les règles (voir section 3.1.1), puis nous montrons comment une règle munie de variables est instanciée en une règle de transformation de graphes (section 3.1.2). Enfin, nous introduisons l'algorithme général d'application d'une règle de transformation avec variables (section 3.1.3).

3.1.1 Règles de transformation avec variables

Dans une règle de transformation avec variables $p : L \leftrightarrow K \hookrightarrow R$, les graphes L , K et R sont des *graphes avec variables*. Intuitivement, un graphe avec variables est un graphe partiellement étiqueté

FIGURE 3.1 – Une règle de transformation p avec variables

– aux sommets et/ou aux arcs – par des variables, ou plus généralement des expressions construites sur des variables. Illustrons ces quelques notions à l’aide de la règle de transformation avec variables $p : L \hookleftarrow K \hookrightarrow R$ de la Fig. 3.1. Dans p , les sommets 1 et 2 sont étiquetés par des constantes qui modélisent des villes. Le sommet 3 de la partie gauche L est étiqueté par une variable x interprétée par une ville. Dans p , les arêtes modélisent la distance qui sépare deux villes. Parmi les étiquettes des arêtes, d est une variable interprétée par une distance. Enfin, on remarque que l’arête qui connecte les sommets 1 et 2 dans R est étiquetée par une expression arithmétique qui sera évaluée au moment de l’instanciation. Notons que les variables que nous prenons pour exemple dans cette section sont appelées *variables attributs* par B. Hoffmann.

Intuitivement, la règle p peut être comprise de la manière suivante. Nous commençons par chercher une ville x entre Paris et Nice telle qu’elle soit située à 950 km de Nice. Il s’agit du motif filtré L . Lorsque p est appliquée à un graphe dans lequel il existe une telle ville x , on supprime x et ses arêtes incidentes puis on ajoute un itinéraire Paris-Nice de distance $950 + d$ avec d la distance qui sépare x de Paris.

3.1.2 Instanciation d’une règle de transformation avec variables

Lors de l’instanciation d’un graphe avec variables, les variables qui étiquettent les sommets et les arcs sont substitués par des valeurs d’interprétation selon une valuation σ , puis les expressions sont évaluées. Le graphe ainsi obtenu possède donc des étiquettes classiques, c’est-à-dire sans variable. On note G^σ l’instanciation d’un graphe avec variables G par une valuation σ . L’instanciation d’une règle de transformation avec variables $p : L \hookleftarrow K \hookrightarrow R$ par une valuation σ est alors la production $p^\sigma : L^\sigma \hookleftarrow K^\sigma \hookrightarrow R^\sigma$.

Un exemple d’instanciation d’une règle de transformation avec variables est donné en Fig. 3.2. La règle de transformation de graphes $p^\sigma : L^\sigma \hookleftarrow K^\sigma \hookrightarrow R^\sigma$ est l’instanciation de la règle p de la Fig. 3.1 par la valuation σ telle que $\sigma(x) = Poitiers$, $\sigma(d) = 350$ et donc $\sigma(950 + d) = 950 + \sigma(d) = 1300$.

3.1.3 Application d’une règle de transformation avec variables

Lors de l’application d’une règle de transformation avec variables $p : L \hookleftarrow K \hookrightarrow R$, on ne peut pas générer l’ensemble des instances de p puisqu’il est non borné en général. Ainsi, le principe général de l’application d’une règle de transformation avec variables réside dans la recherche d’une valuation particulière σ qui permet d’instancier p en une règle de transformation applicable. L’application d’une règle de transformation peut être décrite par le schéma algorithmique suivant.

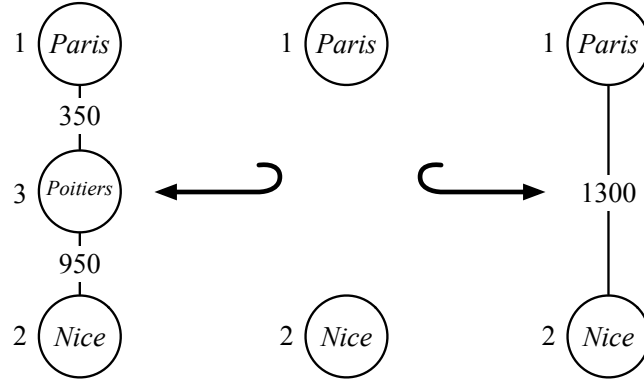


FIGURE 3.2 – L’instanciation p^σ de p par une valuation σ

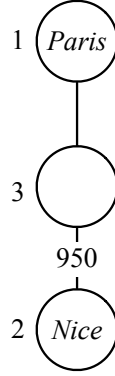
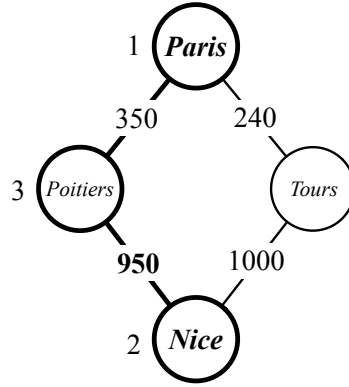


FIGURE 3.3 – \underline{L} , le noyau de L

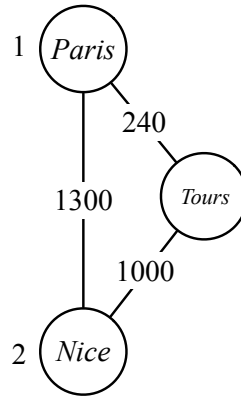
Algorithme. Soit $p : L \hookleftarrow K \hookrightarrow R$ une règle de transformation avec variables et G un graphe. L’application de p au graphe G est obtenu de la manière suivante :

1. Trouver un *morphisme de filtrage de noyau* $\underline{m} : \underline{L} \rightarrow G$. La notion de *noyau d’un graphe avec variables* est illustrée en Fig. 3.3. Intuitivement, il s’agit de supprimer toutes les étiquettes munies de variables. S’il n’existe pas de morphisme de filtrage de noyau \underline{m} , la règle p ne peut pas être appliquée à G .
2. Induire une valuation σ telle que le morphisme de filtrage de noyau \underline{m} s’étende au morphisme de filtrage $m : L^\sigma \rightarrow G$.
3. Construire les graphes K^σ et R^σ afin de construire le graphe H résultat de l’étape de transformation $G \xrightarrow{p^\sigma} H$ (avec $p^\sigma : L^\sigma \hookleftarrow K^\sigma \hookrightarrow R^\sigma$).

Le calcul de σ (étape 2) fait partie intégrante de l’application d’une règle. Afin d’assurer son existence et son unicité pour tout filtrage \underline{m} du noyau, quelques restrictions sur la forme des règles sont généralement considérées. En particulier, la linéarité gauche assure la présence des variables à gauche en une seule occurrence. Associée à l’utilisation des variables à gauche dans des expressions réduites à ces variables, elle garantit l’existence de la valuation σ . De plus, la clôture [Hoffmann 05] assure que toutes les variables de la règle apparaissent à gauche. Ainsi, la valuation σ est unique.

FIGURE 3.4 – Un graphe G

Illustrons l'algorithme d'application d'une règle de transformation avec variables sur un exemple. Pour G , prenons le graphe de la Fig. 3.4 et appliquons-lui la règle p de la Fig. 3.1.

FIGURE 3.5 – Graphe H , résultat de l'étape de transformation directe $G \xrightarrow{p^\sigma, m} H$

À l'étape 1, nous devons trouver un morphisme de filtrage de noyau $\underline{m} : \underline{L} \rightarrow G$. Un tel morphisme existe dans l'exemple. Dans G , la partie en gras représente l'image de \underline{L} par \underline{m} . À l'étape 2, en parcourant séquentiellement les variables qui apparaissent dans L , nous pouvons déduire de \underline{m} la valuation σ telle que $\sigma(x) = Poitiers$ et $\sigma(d) = 350$, \underline{m} s'étend alors au morphisme de filtrage $m : L^\sigma \rightarrow G$. Enfin, à l'étape 4, la valuation σ nous permet de construire les graphes K^σ et R^σ , construisant ainsi la règle de transformation de graphes $p : L^\sigma \hookleftarrow K^\sigma \hookrightarrow R^\sigma$ (voir Fig. 3.2). Le graphe H , résultat de l'étape de transformation directe $G \xrightarrow{p^\sigma, m} H$ est donné en Fig. 3.5.

3.1.4 Variables graphes et clones

Dans la section précédente, nous nous sommes appuyés sur les variables attributs définies par B. Hoffmann afin de présenter l'intuition de la transformation de graphes avec variables. Dans cette section, nous présentons de manière informelle deux autres types de variables introduites dans [Hoffmann 05], proches de nos variables orbites : les *variables clones* et les *variables graphes*.

Les variables clones

Un graphe avec variables clones est un graphe partiellement étiqueté sur $(\Sigma_V \cup (\Sigma_V \times X), \Sigma_E)$ avec Σ_V et Σ_E respectivement des ensembles d'étiquettes de sommets et d'arcs et X un ensemble de variables clones. Intuitivement, une variable clone abstrait un ensemble de sommets.

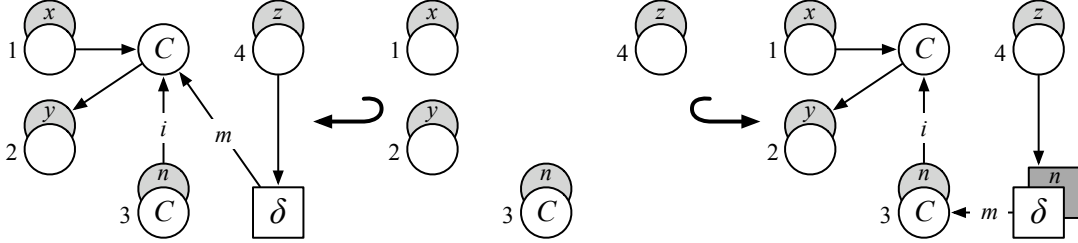


FIGURE 3.6 – Une règle de transformation p avec variables clones

Un exemple de règle avec variables clones $p : L \leftrightarrow K \hookrightarrow R$, avec $\Sigma_V = \{C, \delta\}$, $\Sigma_E = \{i, m\}$ et $X = \{n, x, y, z\}$, est donné en Fig. 3.6. Sur la figure, les variables qui étiquettent les sommets sont représentées sur fond gris. Cet exemple issu de [Hoffmann 05] est inspiré par une méthode de refactorisation de code utilisée en programmation orientée objet, et souvent citée comme exemple d'application de la transformation de graphes [Van Eetvelde 04]. La règle avec variables clones p décrit une transformation d'un programme orienté objet où la définition d'une méthode (représentée par un sommet carré étiqueté par δ) descend depuis une classe mère (représentée par le sommet constant étiqueté par C) vers ses classes filles (représentées par le sommet 3 étiqueté par C et muni d'une variable clone n). Les arcs orientés représentent les liens d'héritage entre les différentes classes.

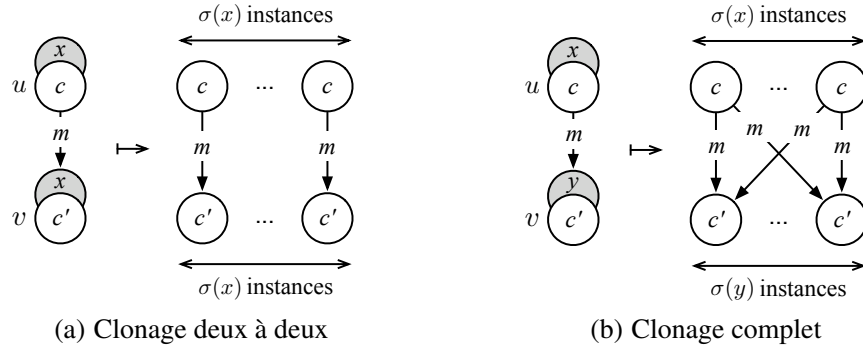


FIGURE 3.7 – La duplication des arcs lors de l'instanciation d'une règle avec variables clones

Dans une règle avec variables clones, les variables clones sont interprétées par un entier qui indique combien de fois chaque clone est dupliqué lors de l'instanciation. L'instanciation d'un graphe avec variables clones est obtenu de la manière suivante :

- un sommet v étiqueté par une variable clone x est dupliqué en $\sigma(x)$ sommets appelés les *instances* de v ;
- lorsqu'un sommet v étiqueté par une variable clone est connecté à un sommet u qui n'est pas muni de variable, les instances de v sont toutes connectées à u ;
- lorsque deux sommets u et v sont étiquetés par une même variable clone x , on effectue un *clonage deux à deux* (voir Fig. 3.7(a)). Les instances de u et de v sont connectées deux à deux ;

- lorsque deux sommets u et v sont étiquetés par deux variables clones distinctes, on effectue un *clonage complet* (voir Fig. 3.7(b)). Chaque instance de u est connectée à toutes les instances de v .

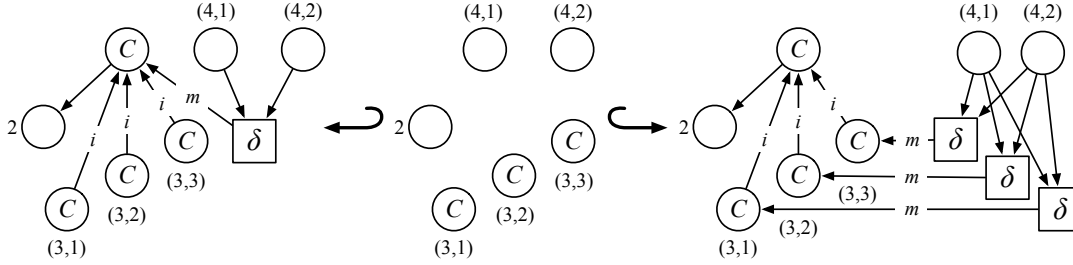


FIGURE 3.8 – L'instanciation p^σ de la règle avec variables clones p par la valuation σ

L'instanciation $p^\sigma = L^\sigma \leftrightarrow K^\sigma \leftrightarrow R^\sigma$ de la règle avec variables clones $p = L \leftrightarrow K \leftrightarrow R$ par la fonction de valuation σ – avec $\sigma(n) = 3$, $\sigma(x) = 0$, $\sigma(y) = 1$ et $\sigma(z) = 2$ – est donnée en Fig. 3.8. Les sommets de p^σ sont nommés par des couples. Par exemple, le sommet $(4, 1)$ de L^σ est la copie 1 du sommet 4 de L . On remarque que le mécanisme de clonage complet est mis en jeu lorsque l'on traite l'arc qui dans R , connecte le sommet 4 au sommet étiqueté par δ . Dans R^σ , toutes les instances du sommet 4 – c'est-à-dire les sommets $(4, 1)$ et $(4, 2)$ de R^σ – sont connectées à toutes les instances du sommet étiqueté par δ .

Le principal avantage des règles avec variables clones réside dans l'automatisation de l'instanciation. En effet, les mécanismes de clonage deux à deux et de clonage complet déterminent automatiquement de quelle manière les instances des clones sont connectées entre elles et avec le contexte. La contrepartie de cette automatisation est la faible expressivité des variables clones. En particulier, les schémas de reconnexion imposés ne nous permettent pas d'utiliser les variables clones pour abstraire les orbites qui paramètrent, par exemple, l'opération de couture des cartes généralisées.

Les variables graphes

Un graphe avec variables graphes est un graphe partiellement étiqueté sur $(\Sigma_V \cup X, \Sigma_E)$ avec Σ_V et Σ_E des ensembles d'étiquettes respectivement de sommets et d'arcs et X un ensemble de variables graphes qui abstraient des sous-graphes. À chaque variable $\delta \in X$ est associé un type $type(\delta) \in \Sigma_E^* \times \Sigma_E^*$ constitué d'un couple de mots qui indique la séquence d'étiquettes des arcs respectivement entrants et sortants de δ . Dans le cas des orbites, les arêtes de connexion d'une orbite à son contexte sont données par le complémentaire du type de l'orbite. Par exemple, dans une 2-G-carte, une demi-face – c'est-à-dire une orbite de type $\langle \alpha_0 \alpha_1 \rangle$ – est connectée par α_2 à une autre demi-face, ou bien à elle-même si elle appartient au bord de l'objet. Notons enfin que dans un graphe avec variables graphes, on impose que les sommets étiquetés par des variables ne soient jamais connectés entre eux afin d'assurer une interface de reconnexion fixe. Les variables graphes ne peuvent donc pas être utilisées pour abstraire des orbites, dont l'interface de reconnexion dépend le plus souvent du nombre brins de l'orbite. De plus, les variables graphes ne nous permettent pas de connecter des orbites entre elles, comme dans l'opération de couture.

La règle p^σ de la Fig. 3.8 peut être vue comme une règle avec variables graphes où les noeuds porteurs de variables graphes sont symbolisés par des carrés. Dans cet exemple, nous avons donc $\Sigma_V = \{C\}$, $\Sigma_E = \{i, m\}$, $X = \{\delta\}$ et $type(\delta) = (\perp\perp, m)$. Le type de δ signifie que chaque occurrence de δ possède deux arcs entrants d'étiquette quelconque et un arc sortant étiqueté par m .

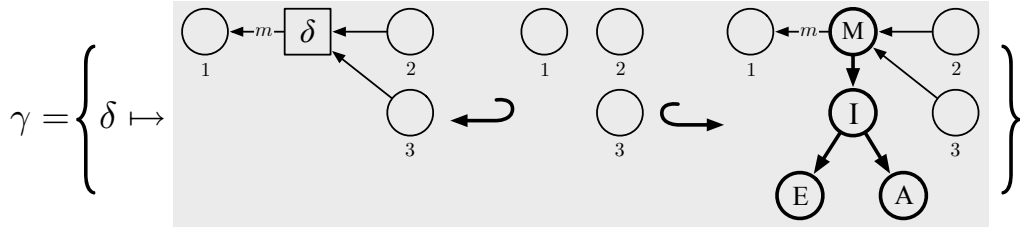


FIGURE 3.9 – Une valuation γ

Dans une règle avec variables graphes, une fonction de valuation associe à chaque variable graphe une règle de transformation de graphes qui indique comment substituer la variable par le graphe qu'elle abstrait. Un exemple de fonction de valuation γ est donné en Fig. 3.9. Remarquons que la valuation γ est elle-même donnée sous la forme d'une règle de transformation de graphes. On remarque que $\gamma(\delta)$ est une règle qui permet de substituer une variable δ par un sous-graphe (représenté en gras sur la figure). Le type $type(\delta) = (\perp\perp, m)$ – donné par la partie gauche de la règle d'instanciation γ – contraint la connexion du sous-graphe au contexte de la variable δ . Intuitivement, lors de l'instanciation de p^σ , la règle $\gamma(\delta)$ est appliquée sur chaque occurrence de δ . L'instanciation $(p^\sigma)^\gamma : (L^\sigma)^\gamma \hookrightarrow (K^\sigma)^\gamma \hookrightarrow (R^\sigma)^\gamma$ de la règle avec variables graphes p^σ par la valuation γ est illustrée en Fig. 3.10.

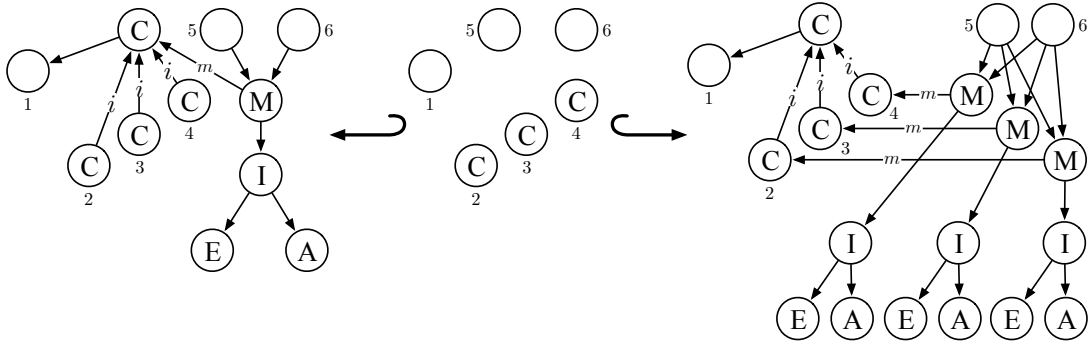


FIGURE 3.10 – L'instanciation $(p^\sigma)^\gamma$ de la règle avec variables graphes p^σ (voir Fig. 3.8) par la valuation γ

Les variables graphes présentent un pouvoir d'expression différent des variables clones. En effet, une variable graphe permet d'abstraire n'importe quel sous-graphe. Ainsi, les variables graphes peuvent être instanciées par des orbites d'une cartes généralisées. Cependant, leur interface de re-connexion est fixe. Nous ne pouvons donc pas utiliser les variables graphes telles quelles pour représenter des orbites liées au reste de la G-carte par tous leurs brins. De plus, la connexion des instances des variables graphes au contexte est exprimée explicitement et de manière *had hoc* dans la fonction de valuation des variables graphes. De même, le fait d'interdire la connexion de deux sommets étiquetés par des variables graphes nous interdit de connecter nos orbites entre elles. Ces limites des règles avec variables graphes ne nous permettent pas d'écrire des règles où, comme dans les règles de 3-couture présentées dans le chapitre précédent, deux orbites $\langle \alpha_0 \alpha_1 \rangle$ sont connectées entre elles (voir les parties droites des règles p_{cou} et p_{cou2} illustrées respectivement en Fig. 2.19 page 45 et Fig. 2.20 page 47).

3.2 Variables orbites

Nous avons vu que les graphes L , K , et R d'une règle de transformation $p : L \leftrightarrow K \hookrightarrow R$ qui traduit l'opération de couture – il en est de même pour l'opération de décousure (voir section 1.4.4) – dépendent du type d'orbite $\langle \alpha_0 \alpha_1 \rangle$ considéré. Nous avons illustré cette observation en considérant une orbite triangulaire et une orbite carrée. Rappelons que nous souhaitons, pour un i donné, pouvoir écrire une règle de transformation générale pour la i -couture (ou la i -décousure) de deux i -cellules le long d'une $(i - 1)$ -cellule commune.

Pour cela, dans cette section, nous proposons de définir des règles de transformation de G-cartes paramétrées par une variable orbite. Nos variables orbites se rapprochent des variables graphes en ce sens qu'elles permettent d'abstraire des sous-graphes, dans notre cas les orbites. Cependant, elles présentent l'avantage d'avoir une interface de re-connexion variable et d'automatiser les traitements de ces connexions, comme dans le cas des variables clones. En particulier, nous n'avons pas besoin d'explicitier la manière dont les orbites substituées sont connectées au contexte, car cette connexion sera construite automatiquement.

En section 3.2.1, nous introduisons nos règles de transformation avec variable orbite. Celles-ci nous permettent de traduire les opérations de couture et de décousure en une unique règle de transformation avec variable. Nous décrirons les mécanismes d'instanciation d'une règle de transformation avec variable orbite en section 3.2.2.

3.2.1 Règles de transformation avec variable orbite

Comme dans le chapitre précédent, nous nous plaçons dans la catégorie des graphes topologiques de dimension n (avec $n \geq 0$), c'est-à-dire dans la catégorie $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ des graphes partiellement étiquetés, où les sommets ne sont pas étiquetés et où les arcs sont partiellement étiquetés sur $\mathcal{A}_n = \{\alpha_0, \dots, \alpha_n\}$.

Dans la suite, nous nous appuyons sur quelques notions classiques ayant trait aux mots et à leurs notations, introduites en annexe A.1. Soit β un mot de \mathcal{A}_n^* . Dans la suite, nous dirons qu'un graphe partiellement étiqueté G est de type $\langle \beta \rangle$ si G est partiellement étiqueté sur $\Sigma = (\emptyset, \Sigma_\beta)$, où Σ_β est l'ensemble des étiquettes du mot β . De même, si β' est un mot de \mathcal{A}_n^* tel que $\Sigma_{\beta'} \subseteq \Sigma_\beta$ et si G est de type $\langle \beta' \rangle$, alors G est également de type $\langle \beta \rangle$. Ainsi, remarquons que toute orbite $\langle \beta \rangle (v)$ d'un graphe topologique de dimension n , est par construction un graphe topologique de dimension n et de type $\langle \beta \rangle$.

Intuitivement, une *variable orbite* x de type $\langle \beta \rangle$, notée $x : \langle \beta \rangle$, sera instanciée par un graphe de type $\langle \beta \rangle$. Plus précisément, $x : \langle \beta \rangle$ sera instanciée par une orbite de type $\langle \beta \rangle$ extraite du graphe G par filtrage lors de l'application d'une règle.

Plus généralement, un *graphe avec variable orbite* est un graphe dont tous les sommets étiquetés sont étiquetés par une même variable orbite typée. Ainsi, les sommets d'un graphe avec variable orbite abstraient tous une même orbite. En d'autres termes, ceci revient à paramétrer globalement le graphe par une unique variable. Ainsi, l'instanciation d'un graphe avec variable par une orbite duplique cette orbite le long du graphe et de manière symétrique le graphe le long de l'orbite. Plus précisément, nous allons définir l'instanciation par un produit du graphe par l'orbite, en construisant automatiquement les connexions entre les différentes copies de l'orbite.

Nos variables orbites nous permettent d'écrire des règles de transformation qui codent les opérations topologiques où une même orbite apparaît plusieurs fois. C'est le cas de l'opération de couture où l'on connecte deux orbites isomorphes. En effet, dans la règle p_{cou} de couture de deux volumes le long d'une face triangulaire (voir Fig. 2.19 page 45) on remarque que l'orbite triangulaire est dupliquée

plusieurs fois. Il en est de même dans la règle p_{cou2} de couture de deux volumes le long d'une face carrée (voir Fig. 2.20 page 47). Nous définissons la notion de graphe avec variable orbite de la manière suivante.

Définition 20 (Graphe avec variable orbite). Soient $n \geq 0$ une dimension et $\langle \beta \rangle$ un type orbite. Un graphe topologique de dimension n avec variable orbite $x : \langle \beta \rangle$ est une graphe G de la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\{x : \langle \beta \rangle\}, \mathcal{A}_n)$.

Par abus, G sera dit graphe avec variable orbite de type $\langle \beta \rangle$.

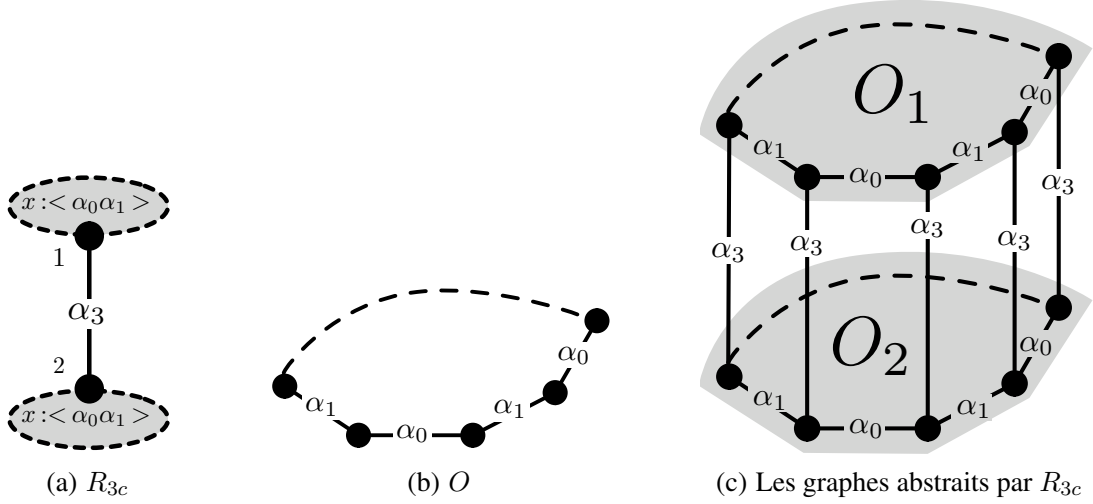


FIGURE 3.11 – Un graphe avec variable orbite R_{3c} et les graphes qu'il abstrait

Un graphe R_{3c} avec une variable orbite $x : \langle \alpha_0 \alpha_1 \rangle$ est donné en exemple sur la Fig. 3.11(a). La variable orbite $x : \langle \alpha_0 \alpha_1 \rangle$ associée aux deux sommets de R_{3c} est globale au graphe. Intuitivement, si on substitue la variable $x : \langle \alpha_0 \alpha_1 \rangle$ de R_{3c} par l'orbite O de la Fig. 3.11(b), on obtient deux copies de O , notées O_1 et O_2 , dont les sommets sont connectés deux à deux par α_3 . Une représentation graphique de ces graphes est proposée en Fig. 3.11(c). Sur la figure, les arêtes en pointillés représentent deux chemins – voir définition 60 page 211 – identiques constitués de sommets et d'arêtes connectés par α_0 et α_1 . Les sommets 1 et 2 de R_{3c} abstraient respectivement les orbites grisées O_1 et O_2 . Remarquons que puisque les sommets étiquetés d'un graphe avec variable orbite sont étiquetés par la même variable, les orbites O_1 et O_2 sont bien identiques.

Comme toujours, le nom x de la variable n'a pas d'autre utilité que de lier les paramètres formels d'une règle générique à leurs valeurs d'application. Dans notre cas, comme les graphes avec variable orbite n'ont qu'une seule variable, tout le graphe sera instancié par une unique orbite. En pratique, le nom de la variable peut donc rester implicite. Dans la suite du chapitre, nous laisserons ainsi la variable orbite implicite, en particulier dans la syntaxe graphique. Par exemple, le graphe R_{3c} de la Fig. 3.11(a), qui porte sur la variable $x : \langle \alpha_0 \alpha_1 \rangle$, se réécrit en la partie droite de la règle p_{3c} de la Fig. 3.12(a) où seul le type orbite $\langle \alpha_0 \alpha_1 \rangle$ est explicité.

Intuitivement, une règle de transformation avec variable orbite est une production $p : L \hookrightarrow K \hookrightarrow R$ où L , K et R sont des graphes qui portent la même variable orbite. Nous définissons la notion de production avec variable orbite de la manière suivante.

Définition 21 (Production avec variable orbite). Soit $n \geq 0$ une dimension et $\langle \beta \rangle$ un type orbite. Une production avec variable orbite $x : \langle \beta \rangle$ est une production $L \leftarrow K \rightarrow R$ dans la catégorie $\mathbb{G}e(\{x : \langle \beta \rangle\}, \mathcal{A}_n)$ où L , K et R sont des graphes avec variable orbite $x : \langle \beta \rangle$.

Par abus, nous dirons que L , K et R sont des graphes avec variable orbite de type $\langle \beta \rangle$. Implicitement, ces graphes portent la même variable orbite de type $\langle \beta \rangle$.

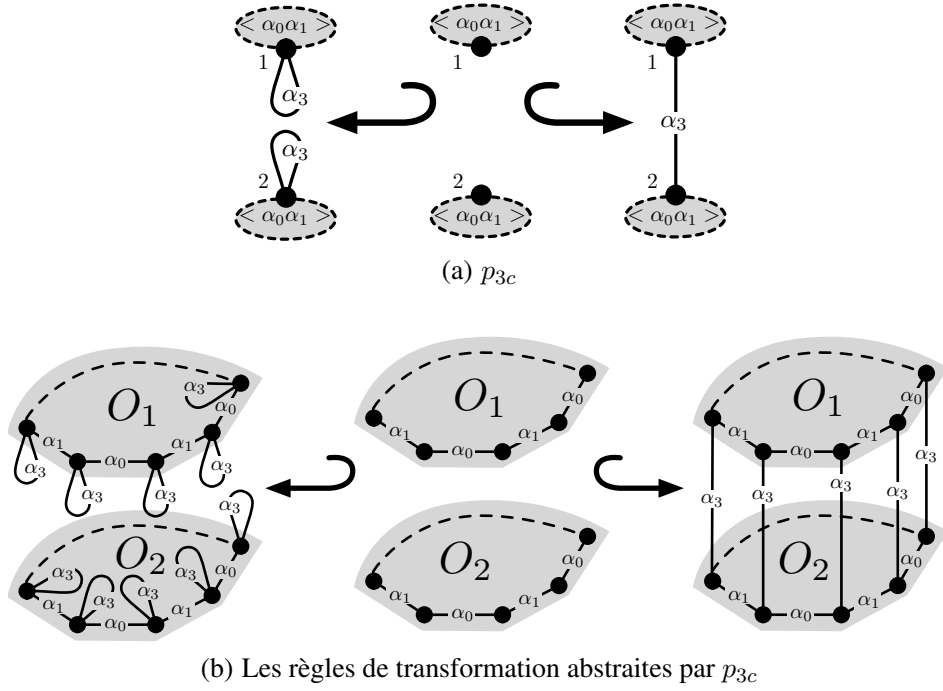


FIGURE 3.12 – La règle de 3-couture p_{3c} et les règles de transformation qu'elle abstrait

Un exemple de règle de transformation de graphes $p_{3c} : L_{3c} \leftrightarrow K_{3c} \hookrightarrow R_{3c}$ avec variable orbite de type $\langle \alpha_0 \alpha_1 \rangle$ est proposé en Fig. 3.12(a). Nous appelons p_{3c} la *règle de 3-couture*. Cette règle abstrait l'ensemble des règles de 3-couture dans une 3-G-carte, celles-ci sont schématisées en Fig. 3.12(b) (où l'on reprend les notations de la Fig. 3.11(c)). La partie gauche L_{3c} représente les graphes constitués de deux orbites de type $\langle \alpha_0 \alpha_1 \rangle$ isomorphes déconnectées où chaque brin possède une boucle incidente étiquetée par α_3 . L'interface K_{3c} représente deux orbites de type $\langle \alpha_0 \alpha_1 \rangle$ isomorphes. Enfin, le graphe R_{3c} est le graphe de la Fig. 3.11(a). Comme les graphes L_{3c} , K_{3c} et R_{3c} portent sur la même variable $x : \langle \alpha_0 \alpha_1 \rangle$, les orbites O_1 et O_2 présentes dans p_{3c} sont toutes identiques.

Rappelons que pour être applicable, une règle de transformation avec variables doit être close (voir section 3.1.3). Dans notre cas, il y a une seule variable orbite par règle, cela revient donc à imposer que cette variable apparaisse dans la partie gauche.

3.2.2 Instanciation des variables orbites

Dans cette section, nous introduisons dans un premier temps quelques notions techniques utiles pour l'instanciation de nos règles avec variable orbite. Nous introduisons ainsi les notions d'*élément neutre de dimension n* et de *saturation d'un graphe* (voir section 3.2.2.1). Dans un second temps, nous utilisons ces deux notions pour définir l'instanciation des variables orbites à l'aide d'un mécanisme de produit, tout d'abord dans les graphes (voir section 3.2.2.2), puis dans les règles de transformation (voir section 3.2.2.3).

3.2.2.1 Élément neutre et saturation

Lors de l'instanciation d'un graphe G avec variable orbite $x : < \beta >$ en un graphe sans variable, on interprète la variable x par une orbite O de type $< \beta >$. Dans la pratique, les occurrences de la variable x sont remplacées par des copies de O . De plus, les arcs qui lient deux sommets v et v' dans G sont dupliqués et lient deux à deux les sommets des deux copies de O associées à v et v' .

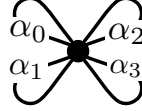
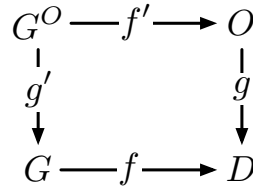


FIGURE 3.13 – L'élément neutre de dimension 3

Dans la pratique, nous calculons l'instanciation G^O d'un graphe G par une orbite O à l'aide d'un produit fibré (voir en annexe A.3.4, la définition 67). Dans le produit fibré (G^O, f', g') illustré par le diagramme suivant,

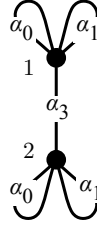


en l'absence d'étiquettes, lorsque le graphe D est réduit à un unique sommet muni d'une unique boucle incidente, le graphe résultat G^O est le produit cartésien des graphes G et O (c'est-à-dire le produit cartésien des sommets et des arcs de G et O). Ce cas particulier de produit fibré est illustré en annexe A.3.4, Fig. A.3(a). Dans notre cas, nous souhaitons effectuer un produit cartésien des sommets de G et O mais une duplication sélective de leurs arcs, en fonction des étiquettes. Pour cela, nous définissons un graphe D qui est muni d'un unique sommet et de $n + 1$ boucles incidentes étiquetées par α_0 à α_n . Nous l'appelons l'élément neutre de dimension n et le notons D_n . Une représentation graphique de D_3 est proposée en Fig. 3.13.

Définition 22 (Élément neutre de dimension n). *L'élément neutre de dimension n est un graphe topologique de dimension n $D_n = (V_{D_n}, E_{D_n}, s_{D_n}, t_{D_n}, l_{V_{D_n}}, l_{E_{D_n}})$ défini par :*

- $V_{D_n} = \{v\}$;
- $E_{D_n} = \{(v, \alpha_i, v) \mid \alpha_i \in \mathcal{A}_n\}$;
- $l_{V_{D_n}} = \perp$;
- s_{D_n}, t_{D_n} et $l_{E_{D_n}}$ sont définis canoniquement.

Dans le cas des cartes généralisées, les étiquettes des arcs de O et G sont distinctes. En effet, lorsque O est de type $< \beta >$, G a tous ses arcs étiquetés sur $\mathcal{A}_n \setminus \Sigma_\beta$. Par conséquent, en raison des morphismes g et f qui préservent les étiquettes des arcs, il n'existe pas de couple d'arcs – l'un pris dans O et l'autre pris dans G – qui se projette sur un même arc dans D_n . Ainsi, le produit fibré précédent – où l'on prend D_n pour D – permet de dupliquer des sommets mais ne permet pas de dupliquer les arcs de O et de G . Afin de dupliquer les arcs de O (respectivement de G), il convient d'ajouter à G (respectivement à O) des boucles incidentes à chaque sommet et portant les étiquettes complémentaires. C'est ce que nous appelons la *saturation*.

FIGURE 3.14 – R_{3c}^{sat}

Intuitivement, la saturation d'un graphe avec variable orbite $x :< \beta >$ revient à ajouter à chaque sommet du graphe des boucles incidentes étiquetées par chaque lettre de β . Par exemple, la saturation du graphe R_{3c} , que l'on note R_{3c}^{sat} est illustrée en Fig. 3.14. Remarquons qu'après saturation, le graphe R_{3c}^{sat} est un graphe topologique étiqueté sur $\mathcal{A}_3 = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$ qui ne contient plus d'occurrence de la variable orbite $x :< \alpha_0 \alpha_1 >$.

Définition 23 (Saturation d'un graphe avec variable orbite). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soit $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ un graphe avec variable orbite $x :< \beta >$.

La saturation du graphe G est le graphe topologique de dimension n noté $G^{sat} = (V_{G^{sat}}, E_{G^{sat}}, s_{G^{sat}}, t_{G^{sat}}, l_{V_{G^{sat}}}, l_{E_{G^{sat}}})$ défini par :

- $V_{G^{sat}} = V_G$;
- $E_{G^{sat}} = E_G \uplus \biguplus_{u \in V_G, i \in [1, |\beta|]} \{e_{u,i}\}$ (où \uplus est l'union disjointe) ;
- pour tout arc $e \in E_G$, c'est-à-dire pour tout arc issu de G :
 - $s_{G^{sat}}(e) = s_G(e)$;
 - $t_{G^{sat}}(e) = t_G(e)$;
 - $l_{E_{G^{sat}}}(e) = l_{E_G}(e)$;
- pour tout arc $e_{u,i} \in E_{G^{sat}} \setminus E_G$, c'est-à-dire pour tout arc issu de la saturation :
 - $s_{G^{sat}}(e_{u,i}) = u$;
 - $t_{G^{sat}}(e_{u,i}) = u$;
 - $l_{E_{G^{sat}}}(e_{u,i}) = \beta(i)$;
- $l_{V_{G^{sat}}}$ est la fonction partout indéfinie.

Reprenons l'exemple de la 3-couture. Lorsque l'on effectue le produit fibré de R_{3c}^{sat} avec l'orbite triangulaire O_t le long de l'élément neutre D_3 (voir Fig. 3.15), nous obtenons bien deux copies de l'orbite O_t dans $R_{3c}^{O_t}$ (une pour chaque sommet 1 et 2 de R_{3c}). En revanche, l'arc α_3 de R_{3c} n'a pas été dupliqué entre les différentes copies de O_t , car O_t ne contient aucun arc étiqueté par α_3 . Nous devons donc saturer O_t par α_3 . Plus généralement, nous devons saturer une orbite d'instanciation de type $< \beta >$ par toutes les étiquettes de $\mathcal{A}_n \setminus \Sigma_\beta$.

Remarquons que si le mot β est vide, la saturation se limite à la suppression de la variable $x :< \beta >$ de l'étiquetage des sommets de G . Remarquons également que la saturation des graphes avec variable orbite préserve l'inclusion des graphes.

Lemme 24. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G et H deux graphes avec la même variable orbite $x :< \beta >$.

Si $G \hookrightarrow H$ alors $G^{sat} \hookrightarrow H^{sat}$.

Démonstration. Considérons les graphes suivants :

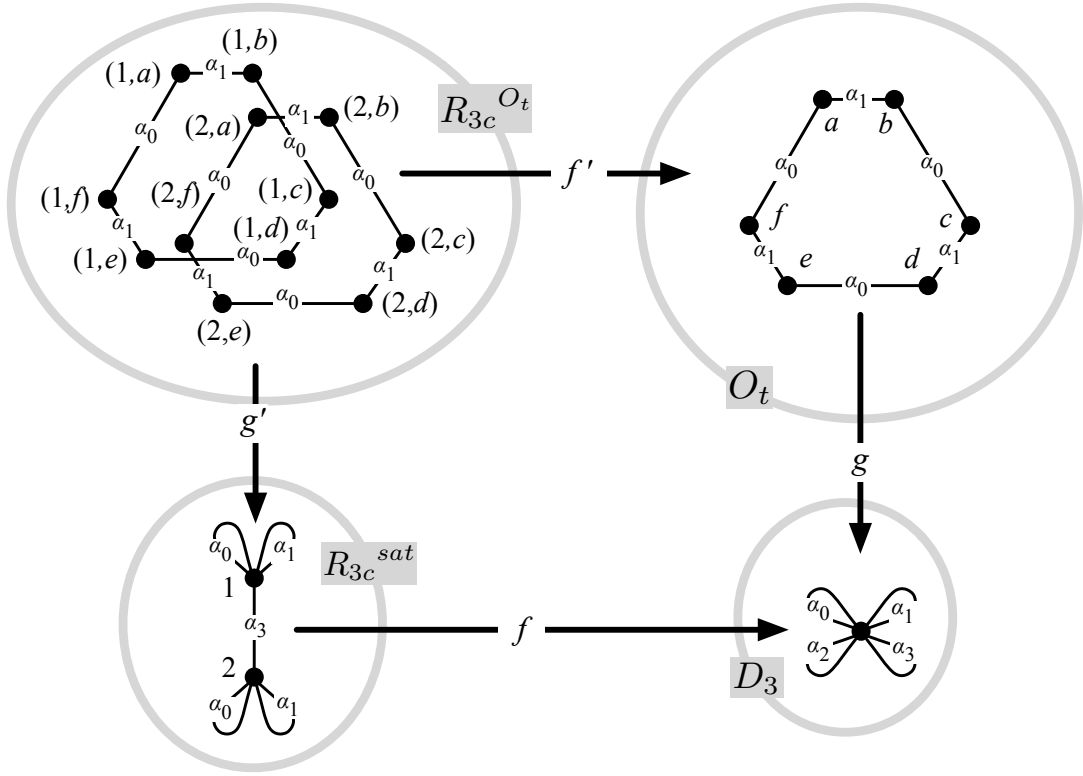


FIGURE 3.15 – Construction des copies de l'orbite O_t

- $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$;
- $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$;
- $G^{sat} = (V_{G^{sat}}, E_{G^{sat}}, s_{G^{sat}}, t_{G^{sat}}, l_{V_{G^{sat}}}, l_{E_{G^{sat}}})$;
- $H^{sat} = (V_{H^{sat}}, E_{H^{sat}}, s_{H^{sat}}, t_{H^{sat}}, l_{V_{H^{sat}}}, l_{E_{H^{sat}}})$.

Soit $f = (f_V, f_E)$ le morphisme d'inclusion de G dans H . Construisons une fonction $f' = (f'_V, f'_E) : G^{sat} \rightarrow H^{sat}$ et montrons qu'il s'agit d'un morphisme d'inclusion.

D'après la définition de la saturation des graphes avec variable orbite $V_G = V_{G^{sat}}$ et $V_H = V_{H^{sat}}$, et donc pour tout sommet $u \in V_{G^{sat}}$ nous posons $f'_V(u) = f_V(u) = u$. De plus, d'après la définition de la saturation des graphes avec variable orbite, les sommets de $V_{G^{sat}}$ et de $V_{H^{sat}}$ ne sont pas étiquetés. La fonction f'_V conserve donc les étiquettes des sommets, partout indéfinies.

De même, d'après la définition de la saturation des graphes avec variable orbite, $E_G \subset E_{G^{sat}}$ et $E_H \subset E_{H^{sat}}$, et donc pour tout arc $e \in E_G$ issu de G nous posons $f'_E(e) = f_E(e) = e$. De plus, comme f est un morphisme de graphes partiellement étiquetés, f préserve les sources, cibles et étiquettes des arcs de E_G . Par conséquent, f' préserve également les sources, cibles et étiquettes des arcs de G^{sat} issus de G .

Considérons à présent les arcs issus des saturations de G et H . Soit u un sommet G . Par hypothèse, u est étiqueté par la variable $x :< \beta >$. Or, par hypothèse, u est également un sommet de H étiqueté par la variable $x :< \beta >$. Donc pour tout indice $i \in [1, |\beta|]$ de β , l'arc $e_{u,i}$ issu de la saturation de G est également un arc issu de la saturation de H car $G \hookrightarrow H$. De plus, d'après la définition de la saturation d'un graphe avec variable orbite, $e_{u,i}$ a pour source et pour cible le sommet u dans G comme dans H . De plus, il est étiqueté par $\beta(i)$ dans G et dans H . Posons $f'_E(e_{u,i}) = e_{u,i}$, la fonction f' préserve donc

les sources, cibles et étiquettes des arcs $e_{u,i}$ ajoutés par saturation.

La fonction f' est donc un morphisme d'inclusion de G^{sat} dans H^{sat} .

□

Comme les orbites ne sont pas des graphes avec variables, nous introduisons une nouvelle définition de la saturation dédiée aux graphes topologiques.

Définition 25 (Saturation d'un graphe topologique). Soient $n \geq 0$ une dimension et $\mathcal{A}_n' \subset \mathcal{A}_n$ un ensemble d'étiquettes. Soit $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ un graphe topologique de dimension n .

La saturation du graphe topologique G par l'ensemble d'étiquettes \mathcal{A}_n' est le graphe topologique de dimension n noté $G^{sat}(\mathcal{A}_n') = (V_{G^{sat}}, E_{G^{sat}}, s_{G^{sat}}, t_{G^{sat}}, l_{V_{G^{sat}}}, l_{E_{G^{sat}}})$ défini par :

- $V_{G^{sat}} = V_G$;
- $E_{G^{sat}} = E_G \uplus \biguplus_{u \in V_G, \alpha_i \in \mathcal{A}_n'} \{e_{u, \alpha_i}\}$ (où \uplus est l'union disjointe) ;
- pour tout arc $e \in E_G$ issu de G :
 - $s_{G^{sat}}(e) = s_G(e)$;
 - $t_{G^{sat}}(e) = t_G(e)$;
 - $l_{E_{G^{sat}}}(e) = l_{E_G}(e)$;
- pour tout arc $e_{u, \alpha_i} \in E_{G^{sat}} \setminus E_G$ issu de la saturation :
 - $s_{G^{sat}}(e_{u, \alpha_i}) = u$;
 - $t_{G^{sat}}(e_{u, \alpha_i}) = u$;
 - $l_{E_{G^{sat}}}(e_{u, \alpha_i}) = \alpha_i$;
- $l_{V_{G^{sat}}}$ est la fonction partout indéfinie.

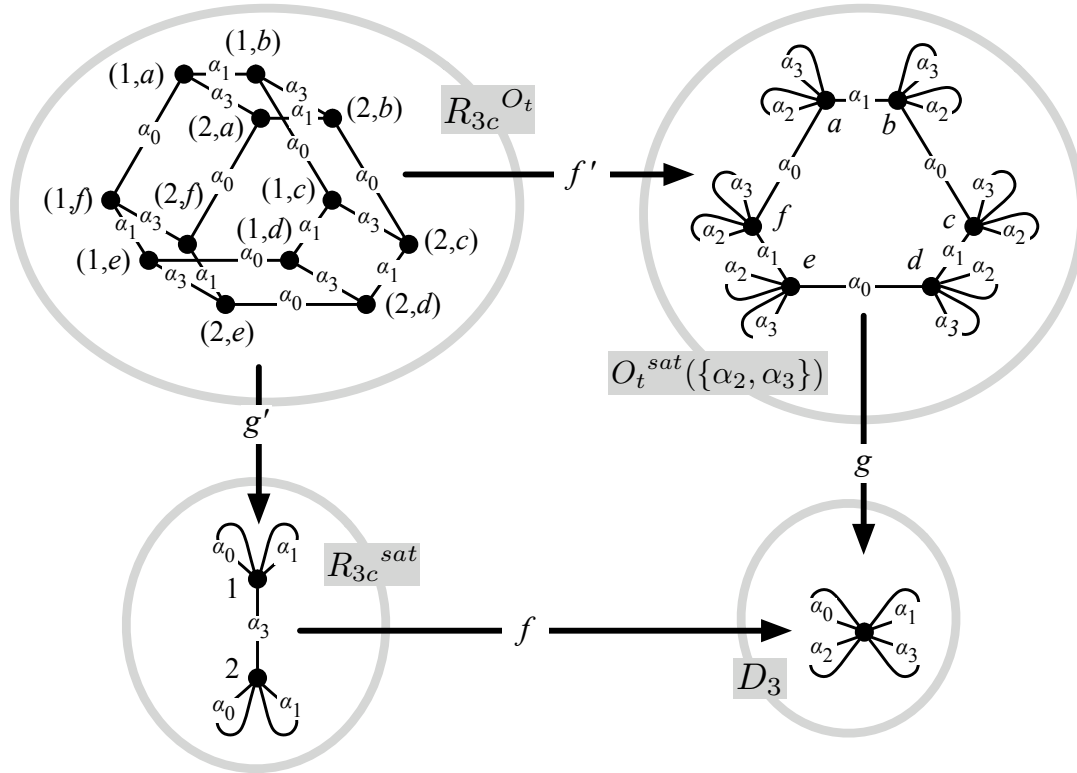


FIGURE 3.16 – Instanciation $R_{3c}^{O_t}$ du graphe R_{3c} par l'orbite triangulaire O_t

Reprenons l'exemple de la 3-couture, Fig. 3.16, où non seulement nous avons saturé R_{3c} en fonction de sa variable de type $\langle \alpha_0 \alpha_1 \rangle$, mais où nous avons également saturé O_t par les étiquettes complémentaires $\{\alpha_2, \alpha_3\}$. Comme le montre la Fig. 3.16, le produit fibré permet comme précédemment de copier l'orbite O_t le long des deux sommets 1 et 2 de R_{3c} . De plus, la saturation de O_t permet de relier deux à deux les sommets des deux copies de O_t , par un arc étiqueté par α_3 copie de l'arc $(1, \alpha_3, 2)$ de R_{3c} .

3.2.2.2 Instanciation des graphes avec variable orbite

Comme nous venons de la voir, l'élément neutre D_n ainsi que la saturation des graphes avec et sans variable orbite nous permet de définir l'instanciation d'un graphe avec variable orbite par produit fibré.

Définition 26 (Instanciation d'un graphe avec variable orbite). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient O une orbite de type $\langle \beta \rangle$ et G un graphe avec variable orbite $x : \langle \beta \rangle$ dont tous les arcs sont étiquetés dans $\mathcal{A}_n \setminus \Sigma_\beta$.

L'instanciation de G par le graphe O , notée G^O , est le graphe topologique de dimension n résultat du produit fibré suivant dans la catégorie $\mathbb{G}e(\emptyset, \mathcal{A}_n)$:

$$\begin{array}{ccc} G^O & \xrightarrow{f'} & O^{sat}(\mathcal{A}_n \setminus \Sigma_\beta) \\ \downarrow g' & & \downarrow g \\ G^{sat} & \xrightarrow{f} & D_n \end{array}$$

avec $f : G^{sat} \rightarrow D_n$ l'unique morphisme de graphes partiellement étiquetés¹ de G^{sat} dans D_n et $g : O^{sat}(\mathcal{A}_n \setminus \Sigma_\beta) \rightarrow D_n$ l'unique morphisme de graphe partiellement étiqueté de $O^{sat}(\mathcal{A}_n \setminus \Sigma_\beta)$ dans D_n .

Nous avons vu en Fig. 3.16 que la définition précédente de l'instanciation produit bien différentes copies de l'orbite reliées par des copies des arcs du graphe avec variable orbite. Naturellement, cette définition peut être appliquée à toute orbite. Ainsi, le graphe R_{3c} – que nous avons instancié par l'orbite triangulaire – peut également être instancié par l'orbite carrée O_c , comme illustré en Fig. 3.17. Nous obtenons alors deux copies de l'orbite carrée reliées deux à deux par des arcs étiquetés par α_3 . Les graphes avec variable orbite sont donc bien des graphes génériques paramétrés par une orbite.

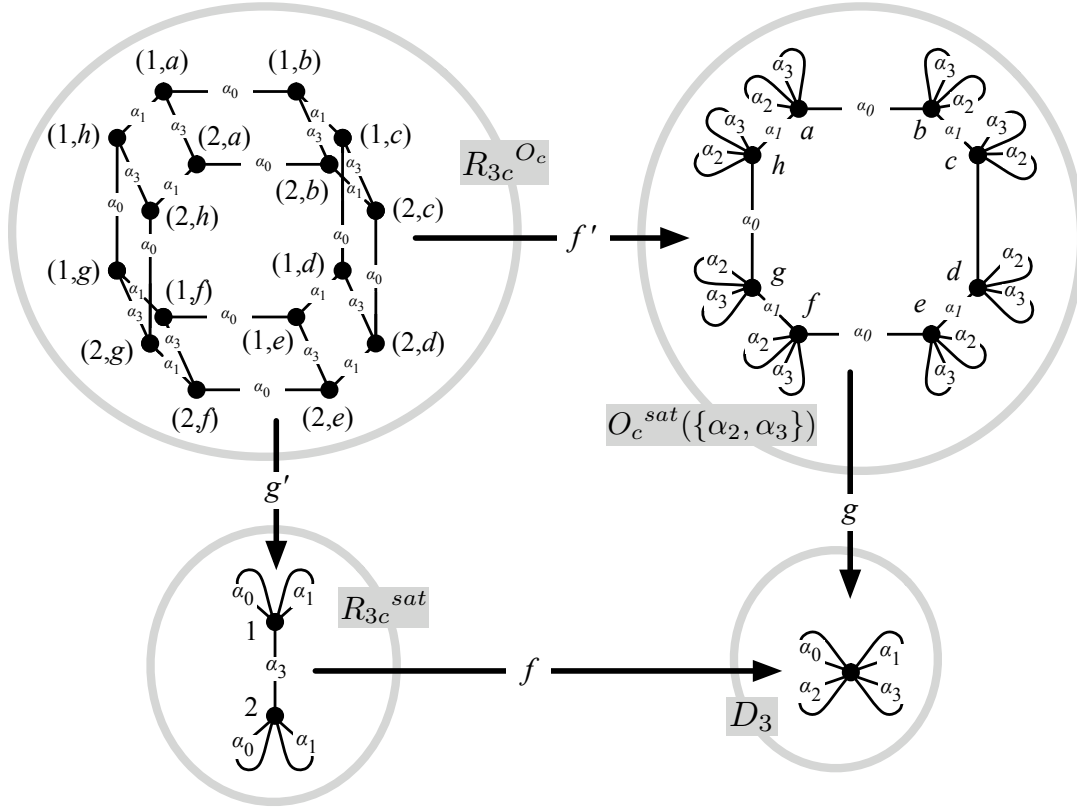
Remarquons enfin que les graphes $R_{3c}^{O_t}$ et $R_{3c}^{O_c}$, respectivement résultats de l'instanciation de R_{3c} par l'orbite triangulaire O_t (voir Fig. 3.16) et par l'orbite carrée O_c (voir Fig. 3.17), sont uniques à isomorphisme près, c'est-à-dire uniques aux noms près des sommets et des arcs.

Lemme 27. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ un graphe avec variable orbite $x : \langle \beta \rangle$ et $O = (V_O, E_O, s_O, t_O, l_{V_O}, l_{E_O})$ une orbite de type $\langle \beta \rangle$.

Soit $I_{G,O} = (V_I, E_I, s_I, t_I, l_{V_I}, l_{E_I})$ le graphe topologique de dimension n suivant :

- $V_I = V_G \times V_O$;
- $E_I = (E_G \times V_O) \uplus (V_G \times E_O)$ où $E_G \times V_O$ est l'ensemble des arcs issus de G et $V_G \times E_O$ est l'ensemble des arcs issus de O ;
- pour tout arc $(e, w) \in E_G \times V_O$ issu de G :

1. Le morphisme f associe à tous les sommets de G^{sat} l'unique sommet v de D_n et associe à tous les arcs étiquetés par α_i de G^{sat} l'arc (v, α_i, v) de D_n

FIGURE 3.17 – Instantiation $R_{3c}^{O_c}$ du graphe R_{3c} par l'orbite carrée O_c

- $s_I((e, w)) = (s_G(e), w)$;
- $t_I((e, w)) = (t_G(e), w)$;
- $l_{E_I}((e, w)) = l_{E_G}(e)$;
- pour tout arc $(v, d) \in V_G \times E_O$ issu de O :
 - $s_I((v, d)) = (v, s_O(d))$;
 - $t_I((v, d)) = (v, t_O(d))$;
 - $l_{E_I}((v, d)) = l_{E_O}(d)$;
- l_{V_I} est la fonction partout indéfinie.

Si tous les arcs de G ont des étiquettes en dehors de Σ_β , alors $I_{G,O}$ est une instantiation de G par O . Nous l'appelons l'instanciation canonique.

Démonstration. Soit $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$ le graphe construit par le produit fibré ci-dessous selon la proposition 68 de l'annexe A.3.4.

$$\begin{array}{ccc}
 H & \xrightarrow{f'} & O^{sat}(\mathcal{A}_n \setminus \Sigma_\beta) \\
 \downarrow g' & & \downarrow g \\
 G^{sat} & \xrightarrow{f} & D_n
 \end{array}$$

Montrons que H est isomorphe à $I_{G,O}$.

Tout d'abord, rappelons que les saturations préservent l'ensemble des sommets. Ainsi, G et G^{sat} ont le même ensemble de sommets V_G . De la même manière, O et $O^{sat}(\mathcal{A}_n \setminus \Sigma_\beta)$ ont le même ensemble de sommets V_O . De plus, D_n a un unique sommet et donc $V_H = V_I = V_G \times V_O$.

Considérons à présent les arcs étiquetés sur Σ_β . Par hypothèse, les arcs de G sont étiquetés en dehors de Σ_β . Par conséquent, tous les arcs de G^{sat} étiquetés sur Σ_β sont issus de la saturation. Ils sont donc de la forme $e_{v,i}$ avec $v \in V_G$ et $i \in [1, |\beta|]$. Par définition, $e_{v,i}$ est étiqueté par $\beta(i)$. De manière symétrique, tout arc de $O^{sat}(\mathcal{A}_n \setminus \Sigma_\beta)$ étiqueté par $\beta(i)$ est un arc de O . Soit d un tel arc. Donc $(e_{v,i}, d)$ est un arc de H de source $(v, s_O(d))$, de cible $(v, t_O(d))$ et d'étiquette $\beta(i)$. Or, (v, d) est un arc de $I_{G,O}$ issu de O et isomorphe à $(e_{v,i}, d)$.

Réciproquement, considérons les arcs étiquetés dans $\mathcal{A}_n \setminus \Sigma_\beta$. Soit e un tel arc de G . Par hypothèse, tous les arcs de O sont étiquetés dans Σ_β . Or, par définition, tout arc de $O^{sat}(\mathcal{A}_n \setminus \Sigma_\beta)$ qui a la même image que e dans D_n a également la même étiquette. Il s'agit donc d'un arc $e_{w,l_{E_G}(e)}$ issu de la saturation. Donc $(e, e_{w,l_{E_G}(e)})$ est un arc de H de source $(s_G(e), w)$, de cible $(t_G(e), w)$ et d'étiquette $l_{E_G}(e)$. Or, (e, w) est un arc de $I_{G,O}$ issu de O isomorphe à $(e, e_{w,l_{E_G}(e)})$.

Ajoutons enfin que $I_{G,O}$ ne possède pas d'autre arc.

Ainsi, $I_{G,O}$ est isomorphe à H et donc est bien une instanciation de G par O . □

Lemme 28. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G et H deux graphes avec variable orbite $x : < \beta >$ tels que $G \hookrightarrow H$ et O une orbite de type $< \beta >$.

Il existe deux instanciations G^O et H^O tels que $G^O \hookrightarrow H^O$.

Démonstration. Montrons que l'instanciation canonique préserve l'inclusion, c'est-à-dire que $I_{G,O} \hookrightarrow I_{H,O}$.

Par construction des instanciations canoniques :

- $V_G \times V_O \subset V_H \times V_O$;
- $E_G \times V_O \subset E_H \times V_O$ et $V_G \times E_O \subset V_H \times E_O$.

Or, d'après le lemme 24, $G^{sat} \hookrightarrow H^{sat}$ et donc les sommets sources, sommets cibles et étiquettes des arcs sont préservés. □

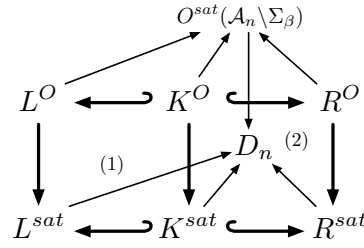
3.2.2.3 Instanciation des règles de transformation avec variable orbite

Intuitivement, l'instanciation d'une production avec variable orbite $p : L \hookrightarrow K \hookrightarrow R$ par une orbite O est une production $p^O : L^O \hookrightarrow K^O \hookrightarrow R^O$ où L^O , K^O et R^O sont respectivement les instanciations de la partie gauche L , de l'interface K et de la partie droite R par l'orbite O .

Définition 29 (Instanciation d'une production avec variable orbite). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient O une orbite de type $< \beta >$ et $p : L \hookrightarrow K \hookrightarrow R$ une production avec variable orbite $x : < \beta >$ dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\{x : < \beta >\}, \mathcal{A}_n \setminus \Sigma_\beta)$.

L'instanciation de p par le graphe O est la production $p^O : L^O \hookrightarrow K^O \hookrightarrow R^O$ telle que les diagrammes (1) et (2) de la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ – représentés en gras sur la figure ci-dessous – commutent² :

2. Remarquons que sur la figure, les carrés disposés en feuillet sont les produits fibrés qui correspondent aux instanciations des graphes L , K et R par le graphe O .



Remarquons que la notion de saturation s'étend aux règles de transformation. La saturation d'une production $p : L \hookrightarrow K \hookrightarrow R$ sur une variable orbite $x : \langle \alpha_0 \alpha_1 \rangle$ est la production $p : L^{sat} \hookrightarrow K^{sat} \hookrightarrow R^{sat}$ où L^{sat} , K^{sat} et R^{sat} sont respectivement les saturations des graphes L , K et R . En Fig. 3.18, nous prenons l'exemple de la saturation p_{3c}^{sat} de la règle de 3-couture sur la variable $x : \langle \alpha_0 \alpha_1 \rangle$. L'existence des morphismes d'inclusion de K_{3c}^{sat} dans L_{3c}^{sat} et de K_{3c}^{sat} dans R_{3c}^{sat} est justifiée par le lemme 24 et l'inclusion de K_{3c} dans L_{3c} et de K_{3c} dans R_{3c} (voir Fig. 3.12(a) page 59). Par conséquent, le lemme 28 assure l'existence des inclusions de K_{3c}^O dans L_{3c}^O et de K_{3c}^O dans R_{3c}^O pour toute orbite O de type $\langle \alpha_0 \alpha_1 \rangle$.

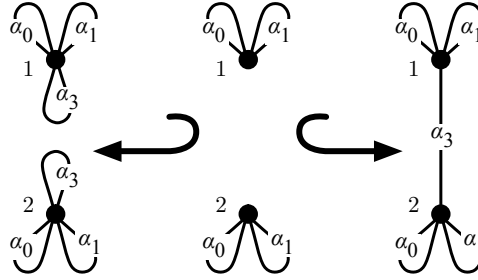


FIGURE 3.18 – Saturation p_{3c}^{sat} de p_{3c} sur la variable $x : \langle \alpha_0 \alpha_1 \rangle$

Dans la définition, l'existence des diagrammes (1) et (2) est la conséquence directe de la préservation de l'inclusion par saturation (voir lemme 24) et par instantiation (voir lemme 28). La commutativité des diagrammes est justifiée par la définition de l'instanciation. En effet, tous les sommets d'une même orbite de type $\langle \beta \rangle$ dans L^O , K^O et R^O ont la même image respectivement dans L^{sat} , K^{sat} et R^{sat} . Il en est de même pour les arcs de l'orbite, pour une étiquette donnée $\alpha_i \in \Sigma_\beta$. Enfin, il en est de même pour les arcs situés en dehors de l'orbite (c'est-à-dire les arcs issus de G), pour une étiquette donnée $\alpha_i \notin \Sigma_\beta$. Remarquons que la commutativité des diagrammes n'est pas remise en cause par l'existence éventuelle dans L , K ou R de deux arcs partageant les mêmes sommets source et cible et la même étiquette. En pratique, ce cas ne se produit pas en raison des contraintes de cohérence des cartes généralisées.

Ainsi, la définition 29 garantit bien l'existence d'une instantiation p^O dans la catégorie $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ pour toute règle p avec variable orbite.

Deux exemples d'instanciation de la règle de 3-couture sont proposés en Fig. 3.19. En Fig. 3.19(a), nousinstancions p_{3c} par l'orbite triangulaire O_t . Remarquons que la règle ainsi obtenue est la règle de couture de deux volumes le long d'une face triangulaire (voir la règle en haut de la Fig. 2.19 page 45). De même, l'instanciation en Fig. 3.19(b) de p_{3c} par l'orbite carrée O_c est la règle de couture de deux volumes le long d'une face carrée (voir Fig. 2.20 page 47).

Ainsi, nous savons à présent exprimer l'opération de 3-couture de deux volumes le long d'une face par une unique règle de transformation avec variable orbite qui ne dépend pas de la face le long de laquelle est effectuée la couture. Par des règles analogues munies d'une unique variable orbite, nous

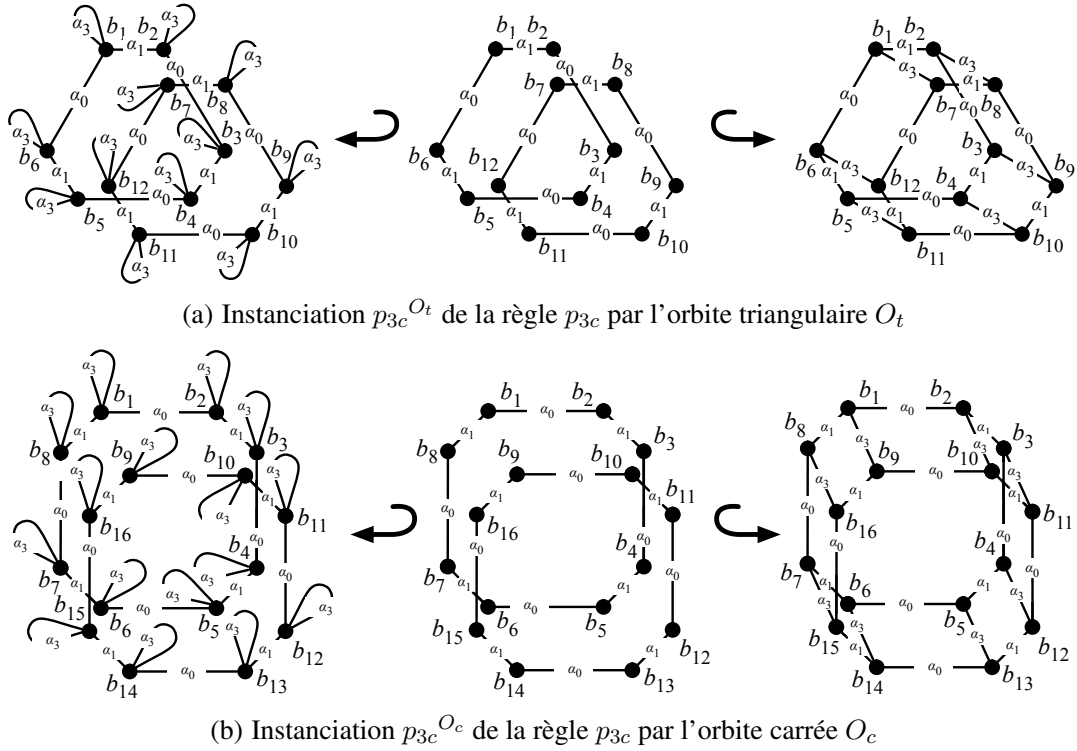


FIGURE 3.19 – Deux instantiations de la règle de transformation avec variable orbite p_{3c}

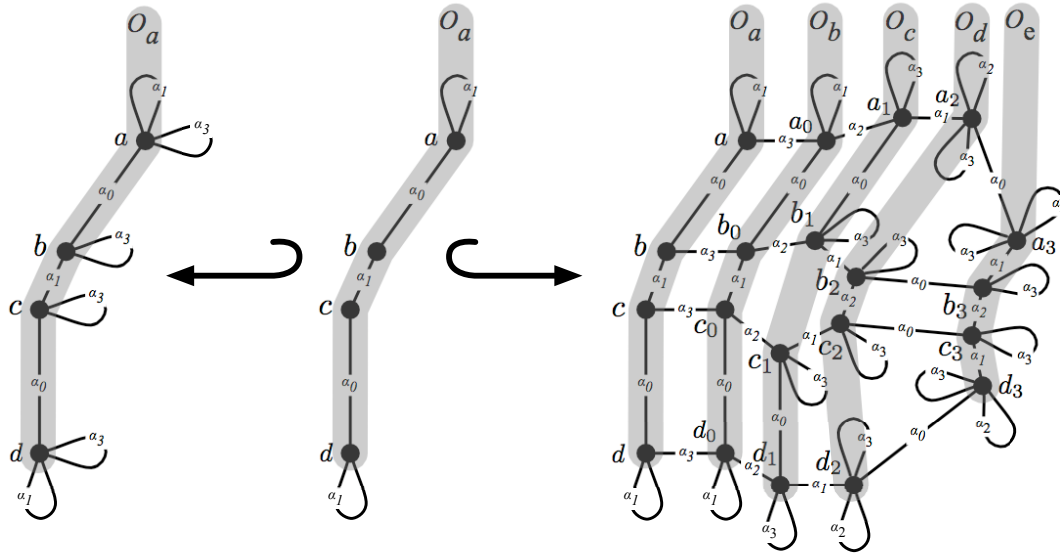
pouvons définir toute couture de deux cellules de dimension i donnée dans une n -G-carte donnée. Dans la section suivante, nous étendons nos règles de transformation avec variable orbite afin de pouvoir traduire les opérations de haut niveau telles que le cône et le chanfreinage (voir section 1.5) en une unique règle.

3.3 Renommages d'orbites

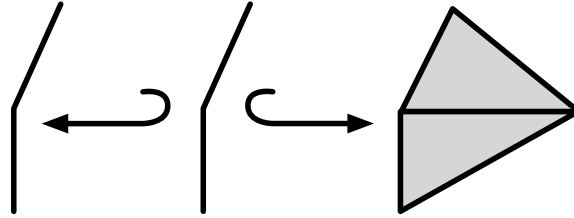
3.3.1 Motivations

Dans les règles de transformation avec variable orbite introduites dans la section précédente, les sommets sont étiquetés par une variable associée à une orbite. Lors de l'instanciation, chaque occurrence de la variable est remplacée par une copie à l'identique de l'orbite associée. Par exemple, lorsque l'on instancie la règle de 3-couture p_{3c} (voir Fig. 3.12(a)) par l'orbite triangulaire O_t (voir Fig. 3.16), les sommets b_1 et b_2 , tous deux étiquetés par la même variable implicite de type $\langle \alpha_0 \alpha_1 \rangle$, sont remplacés par une copie de O_t (voir Fig. 3.19(a)). Ce mécanisme de recopie de l'orbite qui paramètre une règle est suffisant pour traduire les opérations de couture et de décousure en une unique règle de transformation avec variables. Cependant, il ne permet pas de traduire de la même manière les opérations de haut niveau telles que le cône ou le chanfreinage, où l'on manipule des copies d'orbites au renommage et à la suppression près des liaisons (voir section 1.5). Dans cette section, nous introduisons des opérations de *renommage*, qui permettent de traiter le cas des opérations de haut niveau.

Afin d'introduire les renommages, revenons sur l'opération de cône. Prenons l'exemple de l'opération de cône appliquée à un coin de face. Une règle de transformation qui traduit cette opération est proposée en Fig. 3.20(a). L'intuition « géométrique » de la règle est illustrée en Fig. 3.20(b). Dans la partie gauche de la règle, l'orbite $\langle \alpha_0 \alpha_1 \rangle$, notée O_a sur la figure, modélise le coin d'une face que l'on souhaite transformer. On remarque que les brins de O_a ne possèdent pas de brin voisin par α_3 (tous les arcs étiquetés par α_3 sont des boucles), la face est donc au bord de l'objet. Cette condition est imposée par la définition de l'opération de cône (voir définition 11 page 11). Rappelons que l'opération de cône



(a) Règle de cône sur un coin de face



(b) Intuition de la règle

FIGURE 3.20 – Exemple de règle pour l'opération de cône

revient intuitivement à dupliquer l'orbite cible en plusieurs copies identiques au renommage près des liaisons. Ainsi, dans la partie droite de la règle, l'orbite O_a est dupliquée en quatre copies notées O_b , O_c , O_d et O_e . On remarque que l'orbite O_b est une recopie, à l'identique, de l'orbite O_a . Dans O_c , en revanche, les liaisons α_1 de O_a sont supprimées. Dans O_d , les liaisons α_0 sont supprimées et les liaisons α_1 sont renommées en des liaisons α_2 . Enfin, dans l'orbite O_e , les liaisons α_0 et α_1 sont renommées respectivement en liaisons α_1 et α_2 . Intuitivement, les différentes copies d'orbites O_a à O_e sont construites à partir de l'orbite O_a à l'aide de fonction de renommage (et/ou de suppression). Ces transformations sont en réalité définies sur les étiquettes, puisque tous les arcs portant la même étiquette subissent la même transformation.

Intuitivement, nous pouvons traduire l'opération de cône en la règle illustrée en Fig. 3.21(a). Cette règle est paramétrée par une unique variable x de type $\langle \alpha_0 \alpha_1 \rangle$. On remarque que les sommets de la règle sont étiquetés par les fonctions de renommage précédentes appliquées à la variable x .

Ainsi le morphisme strictement alphabétique $\mu_e : \{\alpha_0, \alpha_1\}^* \rightarrow \{\alpha_1, \alpha_2\}^*$, qui renomme l'étiquette α_0 en α_1 et l'étiquette α_1 en α_2 permet de définir la fonction de renommage de graphes $f_e : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle \alpha_1 \alpha_2 \rangle$. Cette fonction transforme tout graphe de type $\langle \alpha_0 \alpha_1 \rangle$ en un graphe de type $\langle \alpha_1 \alpha_2 \rangle$ en renommant chacune de ses étiquettes selon μ_e . En particulier, nous avons $f_e(O_a) = O_e$.

Afin de définir la suppression d'arcs dans des graphes, nous nous donnons une étiquette particulière : « $_$ ». Ainsi, le morphisme strictement alphabétique $\mu_c : \{\alpha_0, \alpha_1\}^* \rightarrow \{\alpha_0, _\}^*$, qui ne modifie pas l'étiquette α_0 mais renomme l'étiquette α_1 en « $_$ », permet de définir la fonction de renommage de graphes

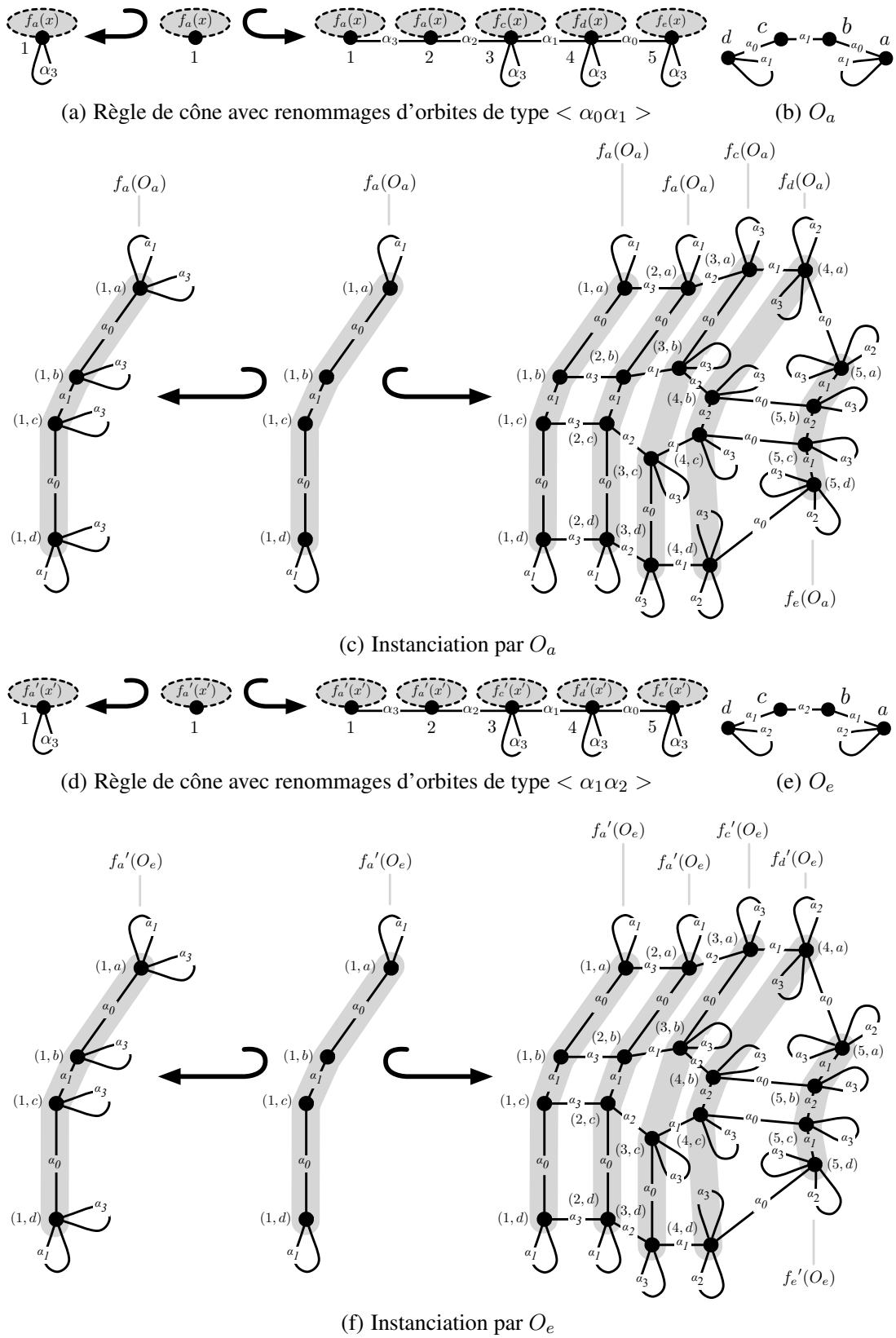


FIGURE 3.21 – Règles de cône avec renommages d'orbites

$f_c : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle \alpha_0 _ \rangle$ (où $\langle \alpha_0 _ \rangle$ est un nouveau type équivalent à $\langle \alpha_0 \rangle$). Cette fonction de renommage supprime tous les arcs étiquetés par α_1 . Dans notre exemple, nous obtenons $f_c(O_a) = O_c$.

Nous pouvons également combiner renommages et suppressions. Par exemple, la fonction $f_d : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle _ \alpha_2 \rangle$, qui transforme l'orbite O_a en O_d , est définie sur le morphisme strictement alphabétique $\mu_d : \{\alpha_0, \alpha_1\}^* \rightarrow \{_, \alpha_2\}$ qui associe à α_0 l'étiquette de suppression « $_$ » et à α_1 l'étiquette de renommage α_2 .

Enfin, O_b est identique à O_a et peut donc être obtenue grâce à la fonction de renommage identité $f_a : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle \alpha_0 \alpha_1 \rangle$ définie sur le morphisme strictement alphabétique identité.

Considérons à présent une orbite O_a de type $\langle \alpha_0 \alpha_1 \rangle$ (voir Fig. 3.21(b)) qui représente le coin d'une face et substituons x par O_a . Nous pouvons interpréter les expressions d'étiquetage afin de retrouver les orbites précédentes : $f_a(O_a) = O_a = O_b$, $f_c(O_a) = O_c$, $f_d(O_a) = O_d$ et $f_e(O_a) = O_e$. Nous allons ainsi pouvoir définir une instanciation avec renommages qui permet d'instancier la règle générique (voir Fig. 3.21(a)) avec l'orbite O_a (voir Fig. 3.21(b)) et d'obtenir la règle de la Fig. 3.21(c).

Remarquons que pour une variable x de type $\langle \beta \rangle$ (dans notre exemple, $\beta = \alpha_0 \alpha_1$), nous utilisons des morphismes de mots μ définis sur Σ_β et des fonctions de renommage f , définies des graphes de type $\langle \beta \rangle$ vers les graphes de type $\langle \mu(\beta) \rangle$. Or, comme μ est un morphisme strictement alphabétique, il est entièrement défini par la donnée de β et de $\mu(\beta)$. Ainsi, μ et f sont directement définis par le profil $\langle \beta \rangle \rightarrow \langle \mu(\beta) \rangle$.

De plus, en pratique, les types de départ des fonctions de renommage sont toujours définis par des mots β dont toutes les lettres sont différentes entre elles et différentes de l'étiquette « $_$ ». Ainsi, pour tout mot γ de $(\mathcal{A}_n \cup \{_\})^*$, il existe un unique morphisme de mot $\mu : \Sigma_\beta^* \rightarrow \Sigma_\gamma^*$ tel que $\mu(\beta) = \gamma$, et il existe une unique fonction de renommage de profil $\langle \beta \rangle \rightarrow \langle \gamma \rangle$. Les fonctions de renommage peuvent donc rester anonymes et être notées par leur seul profil.

Considérons à présent l'orbite O_e de type $\langle \alpha_1 \alpha_2 \rangle$ (voir Fig. 3.21(e)) qui représente un sommet géométrique. Remarquons que la fonction de renommage $f : \langle \alpha_1 \alpha_2 \rangle \rightarrow \langle \alpha_0 \alpha_1 \rangle$ est telle que $f(O_e) = O_a$. Ainsi, à partir de la règle de la Fig. 3.21(a), on peut construire la règle de la Fig. 3.21(d), paramétrée par une variable x' de type $\langle \alpha_1 \alpha_2 \rangle$ et dont les sommets sont étiquetés à l'aide des fonctions de renommage suivantes :

- $f_a' = f_a \circ f : \langle \alpha_1 \alpha_2 \rangle \rightarrow \langle \alpha_0 \alpha_1 \rangle$;
- $f_c' = f_c \circ f : \langle \alpha_1 \alpha_2 \rangle \rightarrow \langle \alpha_0 _ \rangle$;
- $f_d' = f_d \circ f : \langle \alpha_1 \alpha_2 \rangle \rightarrow \langle _ \alpha_2 \rangle$;
- $f_e' = f_e \circ f : \langle \alpha_1 \alpha_2 \rangle \rightarrow \langle \alpha_1 \alpha_2 \rangle$.

L'instanciation de cette nouvelle règle par l'orbite O_e de type $\langle \alpha_1 \alpha_2 \rangle$ (voir Fig. 3.21(e)) est illustrée en Fig. 3.21(f). La règle instanciée est identique à la règle de la Fig. 3.21(c). À présent, les orbites O_a, O_b, O_c, O_d et O_e de la règle de cône sur un coin de face (voir Fig. 3.20(a)) correspondent au renommage par les fonctions f_a', f_c', f_d' et la fonction de renommage identité f_e' .

Le graphe O_d peut donc indifféremment être obtenu par l'instanciation de $x : \langle \alpha_0 \alpha_1 \rangle$ par O_a puis renommage par $f_d : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle _ \alpha_2 \rangle$, ou bien par l'instanciation de $x : \langle \alpha_1 \alpha_2 \rangle$ par O_e puis renommage par $f_d' : \langle \alpha_1 \alpha_2 \rangle \rightarrow \langle _ \alpha_2 \rangle$. La famille de graphes que nous construisons ne dépend donc pas du type de départ de la variable, mais uniquement du type $\langle _ \alpha_2 \rangle$ d'arrivée. Ainsi, dans la suite, f_d et f_d' seront toutes les deux nommées $\langle _ \alpha_2 \rangle$. L'expression $\langle _ \alpha_2 \rangle (x)$ sera alors interprétée par $f_d(x)$ lorsqu'on instanciera x par un graphe de type $\langle \alpha_0 \alpha_1 \rangle$. De manière symétrique, cette même

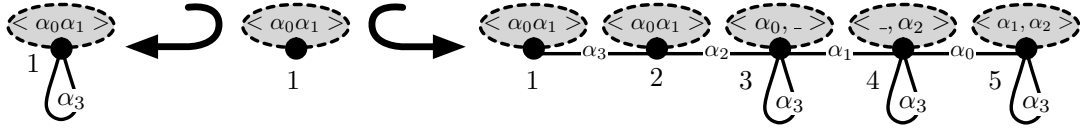


FIGURE 3.22 – p_{cone}

expression sera interprétée par $f_d'(x)$ lorsqu'oninstanciera x par un graphe de type $\langle \alpha_1 \alpha_2 \rangle$. De plus, comme précédemment, puisque la variable x est globale à la règle, nous la laissons implicite. L'étiquette $\langle _ \alpha_2 \rangle$ dénote donc la famille de renommages $\langle _ \alpha_2 \rangle : \langle \beta \rangle \rightarrow \langle _ \alpha_2 \rangle$ où β est un mot de \mathcal{A}_3 , de longueur 2, dont toutes les lettres sont différentes entre elles et différentes de « $_$ ». Cette syntaxe concrète nous permet de réécrire la règle de cône de manière générique – c'est-à-dire indépendamment du type de l'orbite qui instancie x – en la règle p_{cone} de la Fig. 3.22.

L'intérêt des surcharges des opérations de renommage est développé dans les sections 3.3.2 et 3.3.3. Les règles de transformation avec renommages d'orbites ainsi que leur instanciation sont définies respectivement dans les sections 3.3.4 et 3.3.5.

3.3.2 Renommages d'arcs dans les graphes

Cette section – de même que la section suivante – s'appuie sur les notions et notations ayant trait aux mots proposées en annexe A.1.

Afin de manipuler des copies d'orbites isomorphes aux étiquettes des arcs près, nous introduisons la notion de renommage d'arcs dans les graphes. Comme nous l'avons vu, ce renommage d'arcs est directement construit à partir d'un renommage des étiquettes d'arcs.

Définition 30 (fonction de renommage). Soit $n \geq 0$ une dimension. Soient β et γ deux mots de \mathcal{A}_n^* tel qu'il existe un morphisme strictement alphabétique $\mu : \Sigma_\beta^* \rightarrow \Sigma_\gamma^*$ avec $\mu(\beta) = \gamma$

La fonction de renommage $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ définie par μ associe à tout graphe $O = (V, E, s, t, l_V, l_E)$ de type $\langle \beta \rangle$ le graphe $\langle \gamma \rangle(O) = (V, E, s, t, l_V, \mu \circ l_E)$.

Remarquons que le graphe $\langle \gamma \rangle(O)$ est bien de type $\langle \gamma \rangle$ puisque toutes les étiquettes des arcs de O sont renommés sur Σ_γ .

Par définition, $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ est une fonction de renommage s'il existe un morphisme strictement alphabétique $\mu : \Sigma_\beta^* \rightarrow \Sigma_\gamma^*$ tel que $\mu(\beta) = \gamma$, c'est-à-dire si pour tout indice i de β , $\mu(\beta(i)) = \gamma(i)$. Un tel morphisme existe à la condition suivante : si β contient deux lettres identiques aux positions i et j alors γ contient également deux lettres identiques aux mêmes positions. De plus, lorsqu'un tel μ existe, il est complètement défini par les mots β et γ , puisque μ est défini sur les seules lettres Σ_β de β . Cette condition est trivialement vérifiée lorsque toutes les lettres de β sont différentes.

Cela nous permet en pratique, d'utiliser directement les fonctions de renommage sur les orbites avec la garantie que les morphismes de renommage des étiquettes existent. En effet, les orbites sont toujours dénotées par des mots dont toutes les lettres sont différentes (voir définition 3 page 12).

Proposition 31. Soient β , γ et δ trois mots de \mathcal{A}_n^* tel qu'il existe deux morphismes $\mu : \Sigma_\beta^* \rightarrow \Sigma_\gamma^*$ tel que $\mu(\beta) = \gamma$ et $\nu : \Sigma_\gamma^* \rightarrow \Sigma_\delta^*$ tel que $\nu(\gamma) = \delta$.

La composition $\langle \delta \rangle \circ \langle \gamma \rangle$ de $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ et $\langle \delta \rangle : \langle \gamma \rangle \rightarrow \langle \delta \rangle$ est la fonction de renommage $\langle \delta \rangle : \langle \beta \rangle \rightarrow \langle \delta \rangle$ définie par $\nu \circ \mu$.

Démonstration. Par définition, $\nu \circ \mu(\beta) = \delta$, la fonction de renommage $\langle \delta \rangle : \langle \beta \rangle \rightarrow \langle \delta \rangle$ est donc bien définie par $\nu \circ \mu$.

Soit $O = (V, E, s, t, l_V, l_E)$ un graphe de type $\langle \beta \rangle$. Par définition,
 $\langle \delta \rangle : \langle \gamma \rangle \rightarrow \langle \delta \rangle$ ($\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ (O)) = $(V, E, s, t, l_V, \nu \circ (\mu \circ l_E))$ ce qui est naturellement égal à $(V, E, s, t, l_V, (\nu \circ \mu) \circ l_E)$, c'est-à-dire $\langle \delta \rangle : \langle \beta \rangle \rightarrow \langle \delta \rangle$ (O). \square

Cette proposition justifie le fait qu'en pratique, lorsque nous composons deux fonctions de renommage $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ définie par μ et $\langle \delta \rangle : \langle \gamma \rangle \rightarrow \langle \delta \rangle$ définie par ν , nous utilisons directement la fonction de renommage $\langle \delta \rangle : \langle \beta \rangle \rightarrow \langle \delta \rangle$ définie par $\nu \circ \mu$.

De plus, nous avons vu que si $\langle \beta \rangle$ est un type d'orbite, alors toutes les lettres de β sont différentes. Ainsi, pour tout mot γ de \mathcal{A}_n^* de même longueur que β , il existe un unique morphisme strictement alphabétique $\mu : \Sigma_\beta \rightarrow \Sigma_\gamma$ tel que $\mu(\beta) = \gamma$. Il suffit de prendre $\mu(\beta(i)) = \gamma(i)$ pour tout $i \in [1, |\beta|]$. Par conséquent, $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ est bien une fonction de renommage de graphes. En particulier, $\langle \beta \rangle : \langle \beta \rangle \rightarrow \langle \beta \rangle$ existe et est la fonction de renommage identité.

Ainsi, ces deux propositions expliquent que nous ayons introduit un mécanisme de surcharge des noms des fonctions de renommage entre elles et des noms des fonctions de renommage avec les types. Ainsi, dans la syntaxe graphique, la notation $\langle \beta \rangle$ attachée à un sommet peut être interprétée – comme précédemment (voir section 3.2) – par le type $\langle \beta \rangle$ et donc le sommet est directement substitué par une orbite O de type $\langle \beta \rangle$. Ou bien, $\langle \beta \rangle$ peut être interprétée par la fonction de renommage identité $\langle \beta \rangle : \langle \beta \rangle \rightarrow \langle \beta \rangle$ et donc le sommet est également substitué par $\langle \beta \rangle (O) = O$. Enfin, $\langle \beta \rangle$ peut être interprétée par la fonction de renommage $\langle \beta \rangle : \langle \delta \rangle \rightarrow \langle \beta \rangle$ auquel cas le sommet est substitué par le renommage $\langle \beta \rangle (O')$ d'une orbite O' de type $\langle \delta \rangle$.

Dans la suite du chapitre, les graphes et les règles avec renommages d'orbites sont à nouveau paramétrés par une unique variable mais les sommets sont maintenant étiquetés pas les fonctions de renommage.

3.3.3 Renommages et effacements d'arcs dans les graphes

Afin d'étendre le renommage à l'effacement de certains arcs, nous avons enrichi la syntaxe des fonctions de renommage avec une nouvelle étiquette « $_$ » qui dénote les arcs à supprimer. Ceci nous a conduit à étendre l'ensemble des types. Pour tout mot β de $(\mathcal{A}_n \cup \{_\})^*$, $\langle \beta \rangle$ est le type des graphes munis d'arcs étiquetés sur $\Sigma_\beta \setminus \{_\}$. Plus formellement, notons $\mu_ : \Sigma_\beta^* \rightarrow (\Sigma_\beta \setminus \{_\})^*$ le morphisme défini par : $\mu_(_) = \epsilon$ et $\mu_(\alpha_i) = \alpha_i$ pour tout $\alpha_i \in \Sigma_\beta \setminus \{_\}$. Le type $\langle \beta \rangle$ définit le même ensemble de graphes que $\mu_(\beta) \rangle$.

Dans le cas particulier des orbites, l'effacement d'arcs ne produit généralement pas une orbite. Par exemple, en Fig. 3.23, le graphe obtenu en Fig. 3.23(b) par la suppression des arcs α_1 de l'orbite carrée – de type $\langle \alpha_0 \alpha_1 \rangle$ – de la Fig. 3.23(a), est bien un graphe de type $\langle \alpha_0 _ \rangle$ – ou $\langle \alpha_0 \rangle$ – mais n'est pas une orbite. En effet, il est composé de plusieurs composantes connexes.

Nous étendons les fonctions de renommage de la définition 30 aux fonctions de renommage et d'effacement. Ici et dans la suite, le nom « fonction de renommage » désigne cette extension.

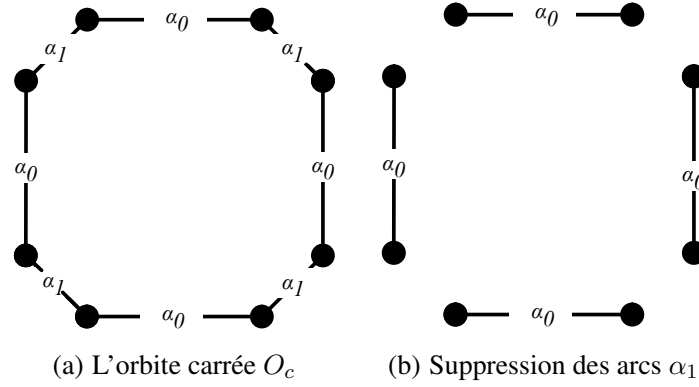


FIGURE 3.23 – Suppression des arcs α_1 dans l'orbite carrée O_c

Définition 32 (fonction de renommage et d'effacement). Soit $n \geq 0$ une dimension. Soient β et γ deux mots de $(\mathcal{A}_n \cup \{_\})^*$ tel qu'il existe un morphisme $\mu : \Sigma_\beta^* \rightarrow \Sigma_\gamma^*$ qui préserve les effacements – c'est-à-dire tel que $\mu(_) = _$ si $_ \in \Sigma_\beta$ – et tel que $\mu(\beta) = \gamma$.

La fonction de renommage $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ définie par μ associée à tout graphe $O = (V_O, E_O, s_O, t_O, l_{V_O}, l_{E_O})$ de type $\langle \beta \rangle$, le graphe $O' = (V_{O'}, E_{O'}, s_{O'}, t_{O'}, l_{V_{O'}}, l_{E_{O'}})$ défini par :

- $V_{O'} = V_O$;
- $E_{O'} = \{e \in E_O \mid \mu(l_{E_O}(e)) \neq _\}$;
- $s_{O'}$, $t_{O'}$ et $l_{E_{O'}}$ sont respectivement les restrictions de s_O , t_O et l_{E_O} sur $E_{O'}$;
- $l_{V_{O'}} = l_{V_O}$.

Notons \mathcal{R}_n l'ensemble des fonctions de renommage de dimension n , c'est-à-dire l'ensemble des fonctions de renommage construites sur des mot de $(\mathcal{A}_n \cup \{_\})^*$.

Comme précédemment, étant donné deux mots β et γ , $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ est une fonction de renommage s'il existe un morphisme $\mu : \Sigma_\beta^* \rightarrow \Sigma_\gamma^*$ qui préserve les effacements et tel que $\mu(\beta) = \gamma$. Ceci est toujours le cas si β est un mot dont toutes les lettres sont différentes de « $_$ » et toutes différentes entre elles. Or en pratique, nous appliquons les fonctions de renommage sur des orbites, dont les types sont toujours exprimés par de tels mots. Ceci nous permet comme précédemment d'utiliser les fonctions de renommage avec l'assurance de leur existence.

Proposition 33. Soit $n \geq 0$ une dimension. Soient β , γ et δ trois mots de $(\mathcal{A}_n \cup \{_\})^*$ tel qu'il existe deux morphismes strictement alphabétiques $\mu : \Sigma_\beta^* \rightarrow \Sigma_\gamma^*$ et $\nu : \Sigma_\gamma^* \rightarrow \Sigma_\delta^*$ qui préservent les effacements et tels que $\mu(\beta) = \gamma$ et $\nu(\gamma) = \delta$.

La composition $\langle \delta \rangle \circ \langle \gamma \rangle$ de $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ et $\langle \delta \rangle : \langle \gamma \rangle \rightarrow \langle \delta \rangle$ est la fonction de renommage $\langle \delta \rangle : \langle \beta \rangle \rightarrow \langle \delta \rangle$ définie par $\nu \circ \mu$.

Démonstration. Par définition, $\nu \circ \mu(\beta) = \delta$ et $\nu \circ \mu$ préserve les effacements, la fonction de renommage $\langle \delta \rangle : \langle \beta \rangle \rightarrow \langle \delta \rangle$ est donc définie par $\nu \circ \mu$.

Soit $O = (V_O, E_O, s_O, t_O, l_{V_O}, l_{E_O})$ un graphe de type $\langle \beta \rangle$. Par définition, $\langle \delta \rangle : \langle \gamma \rangle \rightarrow \langle \delta \rangle$ de $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ (O) = $O' = (V_{O'}, E_{O'}, s_{O'}, t_{O'}, l_{V_{O'}}, l_{E_{O'}})$ avec :

- $V_{O'} = V_O$;
- $E_{O'} = \{e \in E_O \mid \nu(\mu(l_{E_O}(e))) \neq _\}$;
- $s_{O'}$, $t_{O'}$ et $l_{E_{O'}}$ sont respectivement les restrictions de s_O , t_O et l_{E_O} sur $E_{O'}$;
- $l_{V_{O'}} = l_{V_O}$.

Nous avons naturellement $\nu(\mu(l_{E_O}(e))) = (\nu \circ \mu)(l_{E_O}(e))$ et donc $O' = \langle \delta \rangle : \langle \beta \rangle \rightarrow \langle \delta \rangle (O)$. \square

Comme précédemment, cette proposition nous permet en pratique de ne pas composer explicitement les fonctions de renommage, mais d'utiliser directement la fonction composée. Également comme précédemment, si $\langle \beta \rangle$ est un type d'orbites – c'est-à-dire avec β un mot dont toutes les lettres sont différentes entre elles et différentes de « $_$ » – alors pour tout mot $\gamma \in (\mathcal{A}_n \cup \{_\})^*$ de même longueur que β , il existe un unique morphisme strictement alphabétique $\mu : \Sigma_\beta \rightarrow \Sigma_\gamma$ qui préserve les effacements – puisque toutes les lettres de β sont différentes de « $_$ » – tel que $\mu(\beta) = \gamma$. Par conséquent, $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ est bien une fonction de renommage de graphes. La surcharge des fonctions de renommage entre elles reste donc valide lorsqu'il y a des effacements.

3.3.4 Règles de transformation avec renommages d'orbites

Reprenons l'exemple du cône (voir Fig. 3.20). Lors de l'application de la règle, l'orbite O_a de type $\langle \alpha_0 \alpha_1 \rangle$ est filtrée dans la G-carte considérée. Pour construire la partie droite de la règle, les copies $O_a = \langle \alpha_0 \alpha_1 \rangle O_a$ et $O_b = \langle \alpha_0 \alpha_1 \rangle O_a$ sont définies à l'aide du renommage identité $\langle \alpha_0 \alpha_1 \rangle : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle \alpha_0 \alpha_1 \rangle$. Les copies $O_c = \langle \alpha_0 _ \rangle O_a$ et $O_d = \langle _ \alpha_2 \rangle O_a$ sont définies respectivement par les fonctions de renommage (et d'effacement) $\langle \alpha_0 _ \rangle : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle \alpha_0 _ \rangle$ et la fonction de renommage $\langle _ \alpha_2 \rangle : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle _ \alpha_2 \rangle$. Enfin, la copie $O_e = \langle \alpha_1 \alpha_2 \rangle O_a$ est définie par la fonction de renommage $\langle \alpha_1 \alpha_2 \rangle : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle \alpha_1 \alpha_2 \rangle$. La règle de transformation de la Fig. 3.22, reprise en Fig. 3.24, est explicitement paramétrée par les expressions $\langle \alpha_0 \alpha_1 \rangle (x)$, $\langle \alpha_0 _ \rangle (x)$, $\langle _ \alpha_2 \rangle (x)$ et $\langle \alpha_1 \alpha_2 \rangle (x)$, toutes construites sur la même variable x . Lorsque l'on instancie x par l'orbite O_a de type $\langle \alpha_0 \alpha_1 \rangle$, l'ambiguïté sur le type des expressions est levée. Elles peuvent donc être évaluées sur l'instance O_a de x .

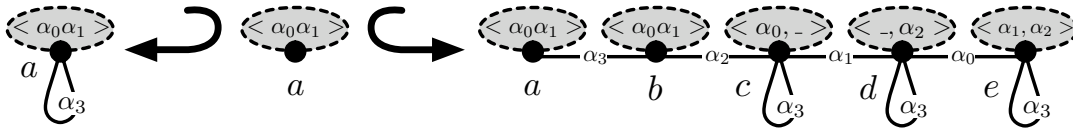


FIGURE 3.24 – Une règle de transformation avec renommages d'orbites

Comme nous venons de le voir sur l'exemple du cône, un graphe avec renommages d'orbite a ses sommets étiquetés par des fonctions de renommage d'orbite. Implicitement, chaque sommet est étiqueté par l'expression formée de l'application de la fonction de renommage sur une variable globale à la règle.

Définition 34 (Graphe avec renommages). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Un graphe avec renommages sur la variable orbite $x : \langle \beta \rangle$ est un graphe partiellement étiqueté de la catégorie $\mathbb{G}e(\mathcal{R}_n(x : \langle \beta \rangle), \mathcal{A}_n)$, où $\mathcal{R}_n(x : \langle \beta \rangle)$ est l'ensemble des expressions de la forme $\langle \gamma \rangle (x)$ avec $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$ une fonction de renommage de \mathcal{R}_n .

Comme nous l'avons vu, la syntaxe des graphes avec renommages laisse la variable – et donc son type – implicite. Un exemple de graphe avec renommages R_{cone} est donné en Fig. 3.25(a), il permet de construire un cône. Dans la pratique, le graphe R_{cone} représente donc toute une famille de graphes avec renommages, pour toute variable orbite $x : \langle \beta \rangle$. Or, tous ces graphes avec renommages abstraient les mêmes graphes (voir Fig. 3.25(b)). En particulier, les instanciations de R_{cone} par les orbites $O_a : \langle \alpha_0 \alpha_1 \rangle$ et $O_e : \langle \alpha_1 \alpha_2 \rangle$ produisent le même graphe. En revanche, il n'est pas possible d'instancier R_{cone} par $O_c : \langle \alpha_0 _ \rangle$ ou $O_d : \langle _ \alpha_2 \rangle$, qui ne sont pas des orbites ($\langle \alpha_0 _ \rangle$ et $\langle _ \alpha_2 \rangle$

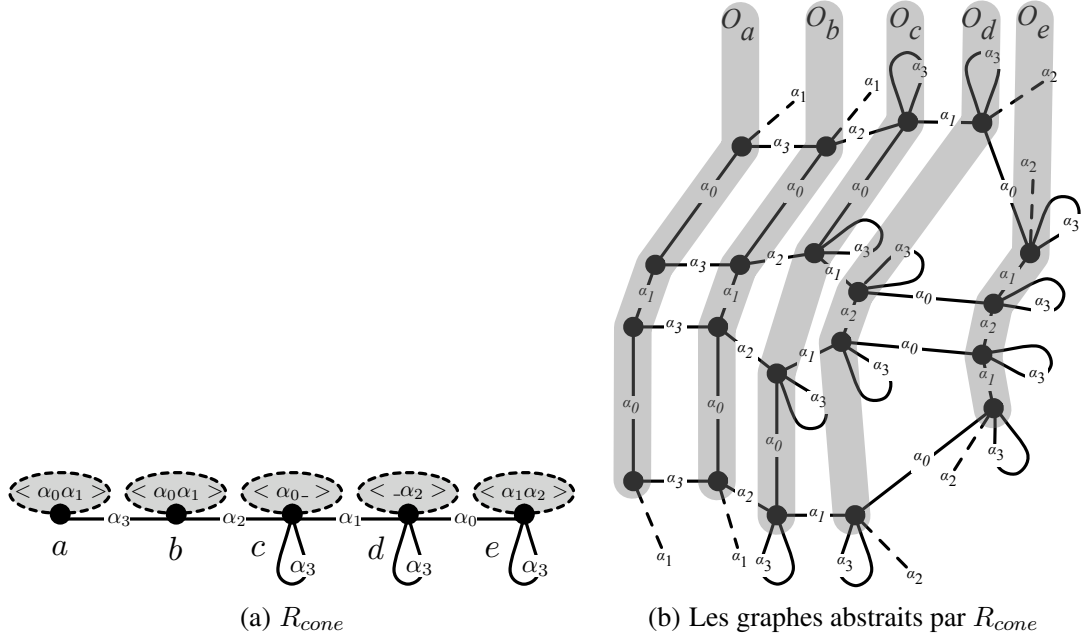


FIGURE 3.25 – Un graphe avec renommages R_{cone} et les graphes qu'il abstrait

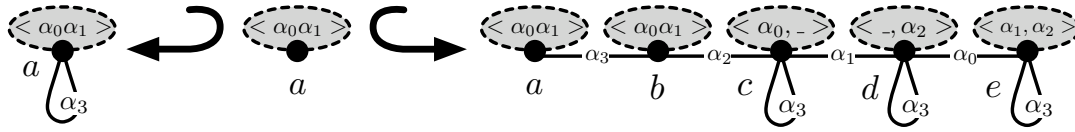
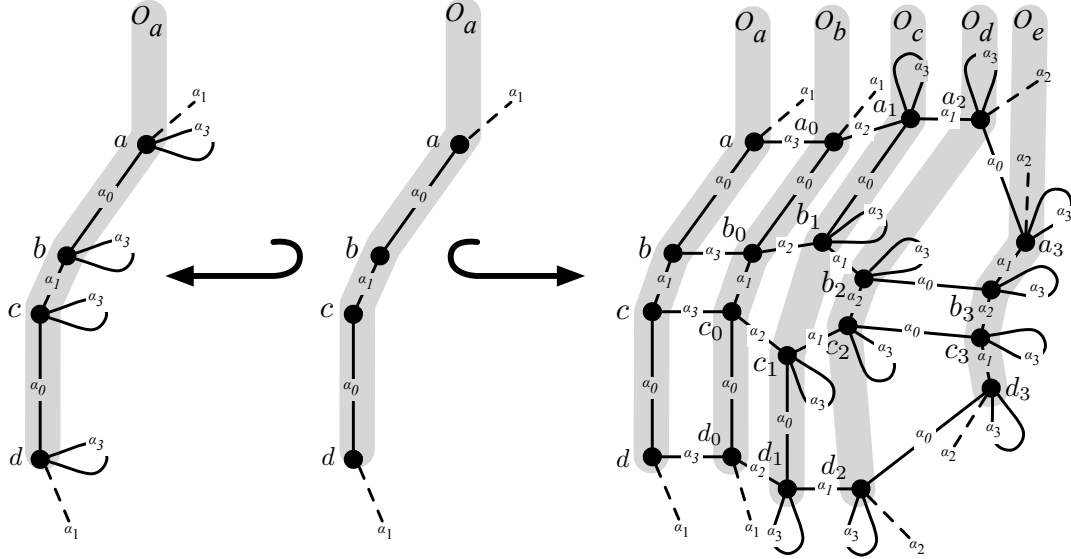
contiennent tous les deux l'étiquette « $_$ »). Par exemple, il n'est pas possible de construire O_a à partir de O_c car il n'existe pas de fonction de renommage de $\langle \alpha_0 - \rangle$ vers $\langle \alpha_0 \alpha_1 \rangle$.

Intuitivement, une règle de transformation avec renommages d'orbite est une production $p : L \hookrightarrow K \hookrightarrow R$ où L , K et R sont des graphes avec renommages. Comme précédemment, la variable orbite est globale à toute la règle.

Définition 35 (Production avec renommages). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Une production avec renommages sur la variable orbite $x : \langle \beta \rangle$ est une production $p : L \hookrightarrow K \hookrightarrow R$ dans la catégorie $\mathbb{G}e(\mathcal{R}_n(x : \langle \beta \rangle), \mathcal{A}_n)$ où L , K et R sont des graphes avec renommages sur la variable $x : \langle \beta \rangle$, tel que L contient au moins un sommet étiqueté par $\langle \beta \rangle (x)$.

Un exemple de règle de transformation de graphes avec termes orbites $p_{cone} : L_{cone} \hookrightarrow K_{cone} \hookrightarrow R_{cone}$ est proposé en Fig. 3.26(a). Nous appelons p_{cone} la règle de cône. Cette règle abstrait l'ensemble des règles qui permettent d'effectuer une opération de cône sur une 2-cellule dans une 3-G-carte. Ces règles sont schématisées en Fig. 3.26(b).

Comme précédemment, la variable x de p_{cone} de même que son type sont implicites. La règle p_{cone} peut donc être instanciée par toute orbite quel que soit son type, en particulier par $O_a : \langle \alpha_0 \alpha_1 \rangle$ et $O_e : \langle \alpha_1 \alpha_2 \rangle$. Cependant, en pratique, l'orbite d'instanciation doit être extraite du graphe G sur lequel on souhaite appliquer la règle p_{cone} . Cette contrainte permet de construire le morphisme de filtrage m de L_{cone} vers G . Intuitivement, si $\underline{m} : L_{cone} \rightarrow G$ est un morphisme de filtrage de noyau de L_{cone} de la règle p_{cone} sur G (voir section 3.1.3), alors p_{cone} sera instanciée par l'orbite $O_a = \underline{m}(a) : \langle \alpha_0 \alpha_1 \rangle$ de G . La construction du morphisme de filtrage $m : O_a \rightarrow G$ est possible si tous les sommets de l'orbite $m(a) : \langle \alpha_0 \alpha_1 \rangle$ de G sont la source d'une boucle étiquetée par α_3 . Dans cet exemple, si G est une carte généralisée, c'est le cas grâce à la contrainte de cycles qui impose l'existence de cycles $\alpha_0 \alpha_3$ et $\alpha_1 \alpha_3$.

(a) p_{cone} (b) Les règles de transformation abstraites par p_{cone} FIGURE 3.26 – La règle de 3-cône p_{cone} et les règles de transformation qu'elle abstrait

3.3.5 Instanciation des renommages d'orbites

Précédemment, nous avons défini l'instanciation en combinant saturation et produit fibré. Cependant, cette définition ne s'applique plus en présence de renommages. En effet, les morphismes de graphes partiellement étiquetés préservent les étiquettes définies (voir en annexe A.2.2, la définition 57). Les fonctions de renommage de graphes ne peuvent donc être directement traduites par des morphismes.

Pour définir une instanciation compatible avec les renommages, nous nous appuyons sur [Habel 02], où Habel et Plump montrent comment renommer des étiquettes de graphes avec des règles de transformation. L'exemple d'une telle règle de transformation est donné en Fig. 3.27. Dans la partie gauche de la règle, les sommets 1 et 2 sont étiquetés respectivement par a et b . Dans la partie droite, l'étiquette du sommet 2 est renommée en c . Pour effectuer ce renommage, il faut que le sommet 2 ne soit pas étiqueté dans l'interface de la règle. Plus généralement, les sommets et arcs renommés par une règle de transformation ne doivent pas être étiquetés dans l'interface.

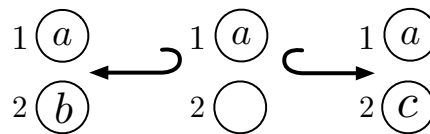


FIGURE 3.27 – Une production qui renomme des étiquettes

Cette construction à deux étages permet de modifier l'étiquette du sommet 2 sans modifier le sommet lui-même. Ainsi, les morphismes de gauche et de droite ajoutent une étiquette différente au sommet 2 qui n'est pas étiqueté dans l'interface de la règle. De même, nous allons définir notre instanciation avec renommages en deux étages. Un premier étage construit – par produit fibré – la structure du graphe. Cette instanciation anonyme a tous ses sommets et arcs définitifs mais aucune étiquette ne leur est attribuée. Le second étage permet alors de rétablir l'étiquetage conformément aux fonctions de renommage.

Dans cette section, nous introduisons la notion de graphe anonyme (voir section 3.3.5.1), qui nous permet de renommer les étiquettes d'un graphe. Ensuite, nous présentons l'instanciation des graphes avec renommages d'orbites (voir section 3.3.5.3), puis au sein des règles de transformation avec renommages d'orbites (voir section 3.3.5.4).

3.3.5.1 Saturation et graphe anonyme

Nous spécialisons la saturation des graphes (voir définition 23) pour les graphes avec renommages d'orbites. Afin que les différentes copies de l'orbite s'effectuent aux renommages et aux suppressions près des arcs, le mécanisme de saturation doit tenir compte des fonctions de renommages associées aux sommets. En particulier, les étiquettes « $_$ » – qui dénotent les suppressions d'arcs – n'ajouteront aucune boucle par saturation.

Définition 36 (Saturation d'un graphe avec renommages d'orbites). *Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soit $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ un graphe avec renommages sur la variable orbite $x : < \beta >$.*

La saturation du graphe G est le graphe topologique de dimension n noté $G^{sat} = (V_{G^{sat}}, E_{G^{sat}}, s_{G^{sat}}, t_{G^{sat}}, l_{V_{G^{sat}}}, l_{E_{G^{sat}}})$ défini par :

- $V_{G^{sat}} = V_G$;
- $E_{G^{sat}} = E_G \uplus \biguplus_{u \in V_G, i \in [1, |\beta|]} \{e_{u,i} \mid l_{V_G}(u) = < \gamma > (x) \text{ et } \gamma(i) \neq _ \}$;
- pour tout arc $e \in E_G$ issu de G , $s_{G^{sat}}(e) = s_G(e)$, $t_{G^{sat}}(e) = t_G(e)$ et $l_{E_{G^{sat}}}(e) = l_{E_G}(e)$;
- pour tout arc $e_{u,i} \in E_{G^{sat}} \setminus E_G$ issu de la saturation, soit $l_{V_G}(u) = < \gamma > (x)$ l'étiquette de u dans G , $s_{G^{sat}}(e_{u,i}) = u$, $t_{G^{sat}}(e_{u,i}) = u$ et $l_{E_{G^{sat}}}(e_{u,i}) = \gamma(i)$;
- $l_{V_{G^{sat}}}$ est la fonction partout indéfinie.

Notons que dans le cas particulier où le graphe G n'aurait aucun renommage effectif (c'est-à-dire lorsque tous les sommets de G sont étiquetés par le renommage identité $< \beta > (x)$), la définition précédente est équivalente à la définition 23 page 61 de la section 3.2.2.1. En effet, dans ce cas, la saturation ajoute à tout sommet u de G toutes les boucles $e_{u,i}$ étiquetées par $\beta(i)$.

Précédemment, nous avons vu que la saturation des graphes avec variable orbite préserve l'inclusion des graphes (voir lemme 24). Cette propriété s'étend à la saturation des graphes avec renommages.

Lemme 37. *Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G et H deux graphes avec renommages sur la variable $x : < \beta >$. Si $G \hookrightarrow H$ alors $G^{sat} \hookrightarrow H^{sat}$.*

Démonstration. Considérons les graphes suivants :

- $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$, $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$;
- $G^{sat} = (V_{G^{sat}}, E_{G^{sat}}, s_{G^{sat}}, t_{G^{sat}}, l_{V_{G^{sat}}}, l_{E_{G^{sat}}})$;
- $H^{sat} = (V_{H^{sat}}, E_{H^{sat}}, s_{H^{sat}}, t_{H^{sat}}, l_{V_{H^{sat}}}, l_{E_{H^{sat}}})$.

Soit $f = (f_V, f_E)$ le morphisme d'inclusion de G dans H . Construisons la fonction $f' = (f'_V, f'_E)$, avec $f'_V : V_{G^{sat}} \rightarrow V_{H^{sat}}$ et $f'_E : E_{G^{sat}} \rightarrow E_{H^{sat}}$, et montrons qu'il s'agit d'un morphisme d'inclusion

de graphes partiellement étiquetés.

D'après la définition de la saturation des graphes avec renommages $V_G = V_{G^{sat}}$ et $V_H = V_{H^{sat}}$, et donc pour tout sommet $u \in V_{G^{sat}}$ nous posons $f'_V(u) = f_V(u) = u$. De plus, d'après la définition de la saturation des graphes avec renommages, les sommets de $V_{G^{sat}}$ et de $V_H = V_{H^{sat}}$ ne sont pas étiquetés. La fonction f'_V conserve donc les étiquettes des sommets.

De même, d'après la définition de la saturation des graphes avec renommages $E_G \subset E_{G^{sat}}$ et $E_H \subset E_{H^{sat}}$, et donc pour tout arc $e \in E_G$ issu de G nous posons $f'_E(e) = f_E(e) = e$. De plus, comme f est un morphisme de graphes partiellement étiquetés, f' conserve les sources, cibles et étiquettes des arcs issus de G .

Considérons à présent les arcs issus des saturations de G et H . Soit u un sommet G étiqueté par un renommage $\langle \gamma \rangle (x)$ et $i \in [1, |\beta|]$. Par hypothèse, u est également un sommet de H et comme f est un morphisme de graphes, u est également étiqueté par le renommage $\langle \gamma \rangle (x)$ dans H . L'arc $e_{u,i}$ issu de la saturation de G est donc également un arc issu de la saturation de H . La fonction f' préserve le sommet u source et cible de $e_{u,i}$ et l'étiquette $\gamma(i)$ de $e_{u,i}$.

Le couple (f'_V, f'_E) est donc un morphisme d'inclusion de G^{sat} dans H^{sat} .

□

Comme nous l'avons vu, le renommage d'étiquettes d'un graphe nécessite de manipuler une version anonyme – c'est-à-dire sans étiquette – du graphe saturé, afin de construire l'instanciation d'un graphe avec renommages en deux étapes.

Définition 38 (Graphe anonyme). Soient $n \geq 0$ une dimension et $G = (V_G, E_G, s_G, t_G, l_V, l_E)$ un graphe topologique de la catégorie $\mathbb{G}e(\emptyset, \mathcal{A}_n)$.

Le graphe anonyme de G est le graphe partiellement étiqueté $G^{ano} = (V_G, E_G, s_G, t_G, l_{V^{ano}}, l_{E^{ano}})$ où $l_{V^{ano}}$ et $l_{E^{ano}}$ sont les fonctions partout indéfinies.

Remarquons que par construction, le graphe G^{ano} est inclus dans G (voir en annexe A.2.2, la définition 59 et la définition 57 respectivement de morphisme d'inclusion de graphe et de morphisme de graphe partiellement étiquetés).

3.3.5.2 Instanciation des graphes avec renommages d'orbites

L'extension de la saturation aux graphes avec renommages ainsi que la notion de graphe anonyme nous permet de construire différentes copies renommées de l'orbite initiale par la double construction de la Fig. 3.28.

Tout d'abord, nous construisons la saturation R_{cone}^{sat} conformément aux différentes fonctions de renommage qui étiquettent les sommets de R_{cone} . Puis nous construisons le graphe $(R_{cone}^{sat})^{ano}$ correspondant en supprimant toutes les étiquettes. Par définition, il existe un morphisme d'inclusion $h : (R_{cone}^{sat})^{ano} \hookrightarrow R_{cone}^{sat}$.

Afin que les copies de l'orbite s'effectuent correctement, il convient de construire f tel que chaque boucle ajoutée par saturation puis anonymée ait la bonne image dans D_n , et ainsi duplique les bons arcs de l'orbite. Dans notre exemple, l'orbite est de type $\langle \alpha_0 \alpha_1 \rangle$. Ainsi, toutes les premières boucles – celles penchées à gauche sur la figure – ont pour image par f la boucle étiquetée par α_0 dans D_n . De même, toutes les secondes boucles – celles penchées à droite – ont pour image la boucle étiquetée par α_2 .

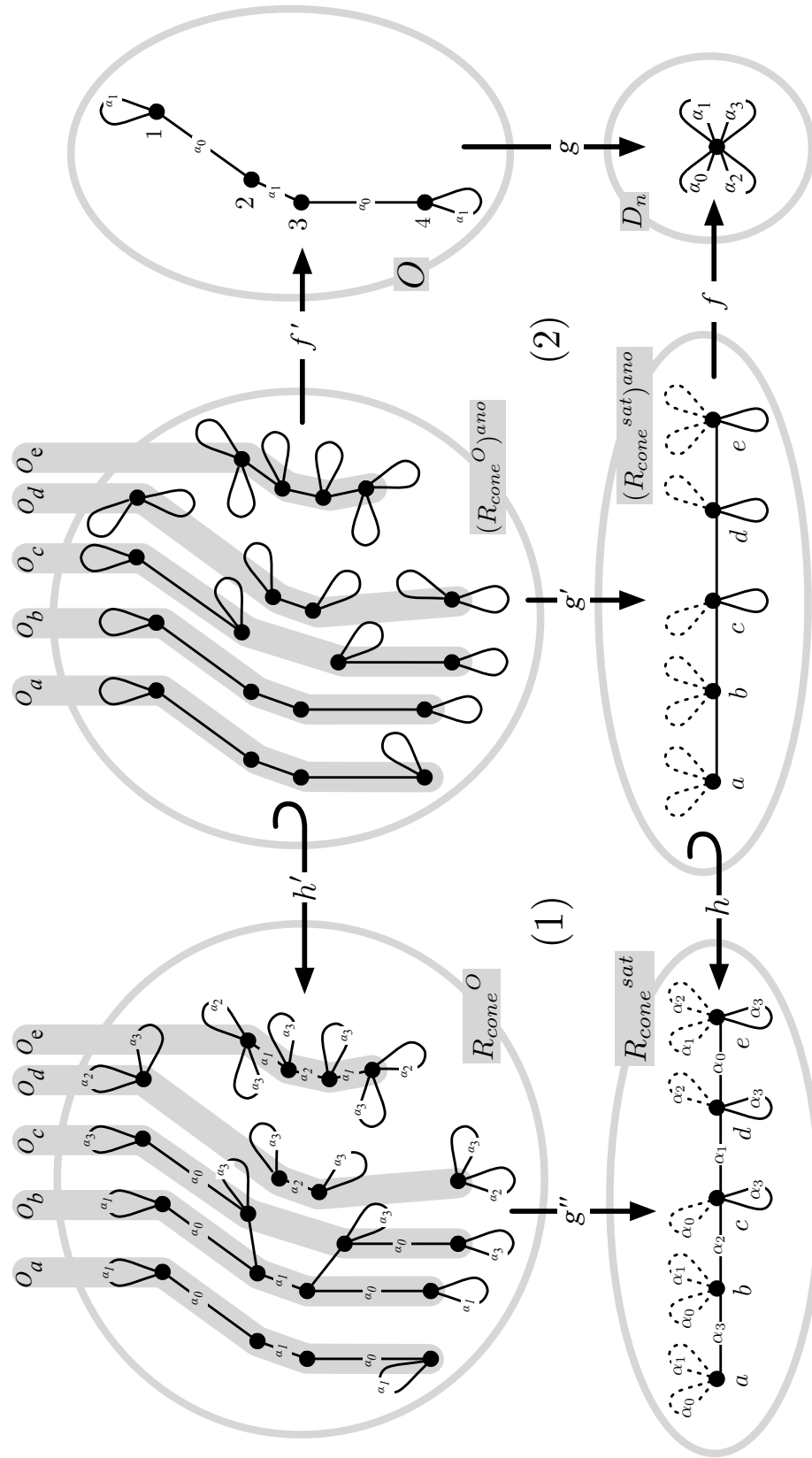


FIGURE 3.28 – Construction des copies renommées de l'orbite O

Lorsque l'on effectue le produit fibré (2), on obtient bien cinq copies de l'orbite O . À ce stade de la construction, les différentes copies sont anonymes mais les suppressions ont déjà été effectuées. En particulier, les sommets c et d – qui n'ont été saturés que par une boucle – ont produit des copies anonymes O_c et O_d qui ne contiennent qu'un arc sur deux.

Le second étage de notre construction – qui correspond au diagramme commutatif (1) – consiste à attribuer leur étiquette aux arcs conformément aux fonctions de renommage. Ainsi, tout arc e construit par produit fibré a une image par g' qui est un arc anonyme issu de la saturation, et une image par h qui porte une étiquette renommée (α_0 dans notre cas). Or, si e est étiqueté dans R_{cone}^O , son image par h' porte nécessairement la même étiquette α_0 . Les orbites O_a et O_e sont donc bien des copies de l'orbite O de départ, sur lesquelles ont été appliqués les renommages idoines.

Comme dans le cas des graphes avec variable orbite dépourvus de renommage, afin de relier les différentes copies de l'orbite entre elles, il faut préalablement saturer cette orbite. Cependant, en raison des renommages, les étiquettes de R_{cone} ne sont pas les étiquettes complémentaires de celles présentes dans l'orbite. Ainsi, dans notre exemple, les arcs de R_{cone} contiennent une étiquette α_0 et une étiquette α_1 . Nous allons donc exploiter notre construction en deux étages pour renommer également les arcs issus de R_{cone} . En particulier, tous ces arcs (représentés en traits pleins sur la figure) doivent avoir par f la même image dans D_n que les arcs issus de la saturation de l'orbite. Pour cela, nous avons choisi d'ajouter une nouvelle étiquette « α » et d'en saturer D_n et l'orbite O (voir Fig. 3.29). Ainsi, tous les arcs issus de R_{cone} ont pour image par f la boucle étiquetée par α ajoutée lors de la saturation. On remarque alors que le produit fibré (2) relie deux à deux les copies anonymes de l'orbite O . Comme pour les arcs issus de O , les liaisons issues de R_{cone} sont ré-étiquetées grâce au diagramme commutatif (1).

Définition 39 (Instanciation d'un graphe avec renommages). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G un graphe avec renommages sur la variable orbite $x :< \beta >$ et O une orbite de type $< \beta >$.

L'instanciation de G par l'orbite O est le graphe topologique de dimension n noté G^O défini par le double diagramme suivant dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n \cup \{\alpha\})$:

$$\begin{array}{ccccc}
 G^O & \xleftarrow{h'} & H & \xrightarrow{f'} & O^{sat}(\{\alpha\}) \\
 \downarrow g'' & & \downarrow g' & & \downarrow g \\
 G^{sat} & \xleftarrow{h} & (G^{sat})^{ano} & \xrightarrow{f} & (D_n)^{sat}(\{\alpha\})
 \end{array}
 \begin{array}{ccc}
 & (1) & (2) \\
 & &
 \end{array}$$

tel que :

- le graphe topologique de dimension n $(G^O)^{ano}$ est le résultat du produit fibré (2) avec :
 - $D_n^{sat}(\{\alpha\})$ est la saturation de l'élément neutre de dimension n par $\{\alpha\}$;
 - $O^{sat}(\{\alpha\})$ est la saturation de l'orbite O par $\{\alpha\}$;
 - $(G^{sat})^{ano}$ est le graphe anonyme de la saturation de G ;
 - $g : O^{sat}(\{\alpha\}) \rightarrow D_n^{sat}(\{\alpha\})$ est l'unique morphisme de graphe partiellement étiqueté de $O^{sat}(\{\alpha\})$ dans $D_n^{sat}(\{\alpha\})$;
 - $f = (f_V, f_E) : (G^{sat}(\Sigma_\beta))^{ano} \rightarrow D_n^{sat}(\{\alpha\})$ est l'unique morphisme tel que :
 - tout arc e issu de G a pour image l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par α ;
 - tout arc $e_{u,i}$ issu de la saturation de G a pour image l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par $\beta(i)$;
- le graphe G^O est l'unique graphe tel que le diagramme (1) commute et dont tous les arcs sont étiquetés. Plus précisément, pour tout arc e de G^O , si l'étiquette de e n'est pas définie dans G^O , alors l'étiquette de $g''(e)$ n'est pas non plus définie dans G^{sat} .

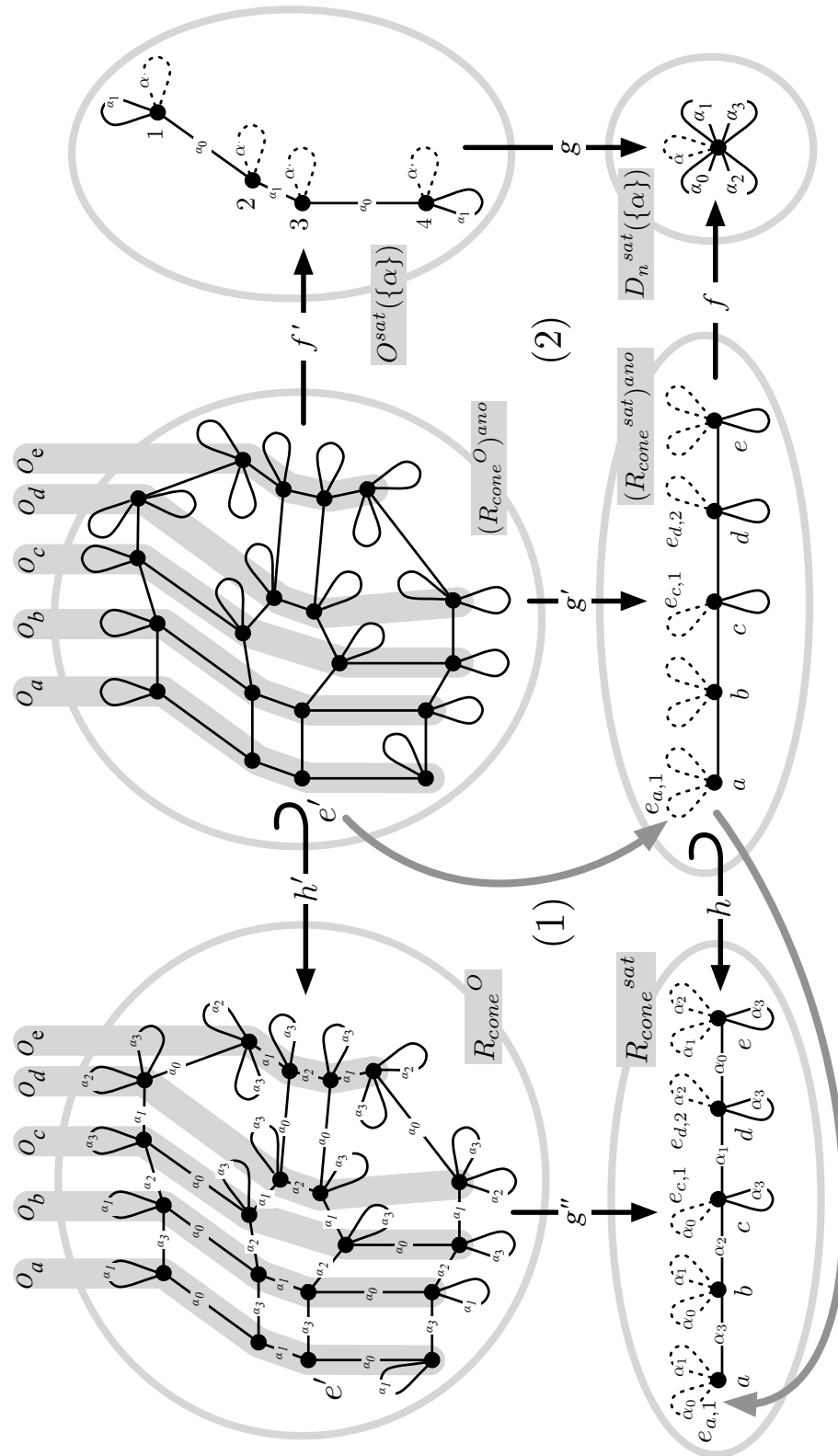


FIGURE 3.29 – Instantiation R_{cone}^O du graphe avec renommages R_{cone} par une orbite O

Remarquons que H est bien le graphe anonyme de G^O . En effet, les ensembles de sommets et d'arcs de G^O et H sont identiques et les sommets et arcs de H ne sont pas étiquetés. Dans la suite, nous noterons directement $(G^O)^{ano}$ pour H .

Commentons la définition en instanciant la partie droite R_{cone} de la règle de 3-cône (voir Fig. 3.26(a) page 77). Rappelons que R_{cone} est un graphe avec renommages de dimension n et de longueur 2. L'instanciation R_{cone}^O de R_{cone} par une orbite O est illustrée en Fig. 3.29. L'orbite O considérée modélise un coin de face, elle correspond à la partie dessinée en traits pleins dans la graphe $O^{sat}(\{\alpha\})$. Afin d'alléger la figure, nous ne nommons pas les sommets des graphes R_{cone}^O et $(R_{cone}^O)^{ano}$.

Intéressons-nous dans un premier temps à la construction du graphe $(R_{cone}^O)^{ano}$ par le produit fibré (2). On remarque que ce graphe est analogue à la partie droite de la règle de cône sur un coin de face (voir Fig. 3.20(a) page 69). En particulier, on retrouve des orbites analogues aux orbites $O_a = \langle \alpha_0 \alpha_1 \rangle (O)$ à $O_e = \langle \alpha_1 \alpha_2 \rangle (O)$. À nouveau, elles correspondent respectivement aux sommets a à e de R_{cone} , étiquetés par les renommages $\langle \alpha_0 \alpha_1 \rangle$ et $\langle \alpha_1 \alpha_2 \rangle$. Toutefois, dans $(R_{cone}^O)^{ano}$, les arcs n'ont pas d'étiquette. De plus, on remarque que les arcs supprimés par renommage dans les orbites O_c et O_d de la règle de cône sur un coin de face sont déjà supprimés dans $(R_{cone}^O)^{ano}$.

Comme dans le produit fibré qui construit l'instanciation des graphes avec variable orbite (voir définition 26 page 64), la construction de $(R_{cone}^O)^{ano}$ nécessite de saturer l'orbite O ainsi que le graphe avec renommages R_{cone} . Sur la Fig. 3.29, les arcs ajoutés lors de la saturation figurent en pointillés. Remarquons que la saturation de R_{cone} tient compte des suppressions d'arcs des renommages $\langle \alpha_0 _ \rangle$ et $\langle _ \alpha_2 \rangle$ qui étiquettent c et d dans R_{cone} (voir définition 36 de la saturation des graphes avec renommages). Ainsi, dans R_{cone}^{sat} , une seule boucle incidente est ajoutée aux sommets c et d lors de la saturation : respectivement les boucles $e_{c,1}$ et $e_{d,2}$. Dans le produit fibré (2), on ne manipule pas directement R_{cone}^{sat} , mais son graphe anonyme $(R_{cone}^{sat})^{ano}$. Ainsi, comme g' est un morphisme de graphes, le fait que $(R_{cone}^{sat})^{ano}$ ne porte pas d'étiquette implique que le graphe $(R_{cone}^O)^{ano}$ résultat du produit fibré ne porte pas non plus d'étiquette. Comme nous l'avons vu dans la section 3.3.5.1, la construction du graphe anonyme $(R_{cone}^O)^{ano}$ nous permet de mettre en œuvre les renommages de nos orbites.

Dans le produit fibré (2), comme dans le cas des graphes avec variable orbite, nous utilisons l'élément neutre de dimension n que nous saturons à présent par une nouvelle étiquette d'arc α . Le morphisme f associe les sommets de $(R_{cone}^{sat})^{ano}$ à l'unique sommet de $D_n^{sat}(\{\alpha\})$. Il associe également les arcs issus de R_{cone} – ceux représentés en traits pleins dans $(R_{cone}^{sat}(\{\alpha_0, \alpha_1\}))^{ano}$ – à la boucle α de $D_n^{sat}(\{\alpha\})$. Enfin, les arcs issus de la saturation sont associés aux boucles étiquetées sur \mathcal{A}_n de $D_n^{sat}(\{\alpha\})$. Par exemple, l'image de la boucle $e_{c,1}$ par f est l'unique boucle étiquetée par $\beta(1)$ de $D_n^{sat}(\{\alpha\})$. De même, l'image de la boucle $e_{d,2}$ est la boucle étiquetée par $\beta(2)$.

La nouvelle étiquette α introduite dans la définition permet de dupliquer les arcs de R_{cone} le long de l'orbite. Précédemment, lors de l'instanciation des graphes avec variable orbite (voir section 3.2.2.2), les étiquettes d'arc des graphes avec variable orbite étaient distinctes de celles de l'orbite. Ainsi, la saturation de l'orbite par les étiquettes complémentaires suffisait à construire les connexions lors de l'instanciation. À présent, les graphes avec renommages peuvent avoir des arcs étiquetés par des étiquettes présentes dans l'orbite. Dans notre exemple, c'est le cas des arcs (c, α_1, d) et (d, α_0, e) de R_{cone} .

Rappelons que la saturation de O nous permet de dupliquer les arcs de R_{cone} afin de connecter entre elles les copies des orbites (voir section 3.2.2). Pour cela, les boucles issues de la saturation de O et les arcs de R_{cone} – qui figurent en traits pleins dans $(R_{cone}^{sat}(\{\alpha_0, \alpha_1\}))^{ano}$ – doivent être projetés sur le même arc dans $D_n^{sat}(\{\alpha\})$. Nous avons vu que les arcs issus de R_{cone} sont projetés sur la boucle α par f . Les boucles qui saturent O doivent donc être projetées sur la boucle α par g . Nous choisissons alors de saturer l'orbite O par l'étiquette α . De cette manière, le morphisme G est l'unique morphisme qui va

de $O^{sat}(\{\alpha\})$ dans $D_n^{sat}(\{\alpha\})$.

À présent que nous avons construit une instanciation anonyme de R_{cone} par l'orbite O , nous devons ajouter les étiquettes des arcs en fonction des opérations de renommage qui étiquettent les sommets de R_{cone} . Pour cela, intéressons-nous au diagramme (1) de la Fig. 3.29.

D'après la définition de l'instanciation des graphes avec renommages, R_{cone}^{ano} est bien le graphe anonyme du graphe résultat de l'instanciation R_{cone} . Ainsi, d'après la définition des graphes anonymes (voir définition 38 page 79), il existe un morphisme d'inclusion de R_{cone}^{ano} dans R_{cone} . En particulier, les ensembles d'arcs de R_{cone}^{ano} et R_{cone} sont égaux. Considérons un arc e' du graphe R_{cone}^{ano} . L'étiquette de l'arc e' dans R_{cone} est obtenue à l'aide des morphismes g' et h . En suivant la flèche grise qui part de e' nous arrivons sur l'arc anonyme $e_{a,1}$, puis en suivant la seconde flèche grise, nous arrivons en R_{cone}^{ren} sur la boucle (a, α_0, a) étiqueté par α_0 . L'arc e' est donc étiqueté par α_0 dans R_{cone}^O . Remarquons enfin que g'' est bien un morphisme de graphe. En particulier, l'étiquette α_0 de la boucle (a, α_0, a) de R_{cone}^{sat} est bien la même que l'étiquette de son antécédent e' par g'' .

3.3.5.3 Équivalence des graphes avec renommages et des graphes avec variable orbite

Comme nous l'avons déjà dit, la syntaxe graphique exploite la surcharge et le fait que la variable orbite soit implicite. Ainsi, un graphe dont tous les sommets sont étiquetés par la même étiquette $< \beta >$ peut être aussi bien interprété par un graphe avec variable orbite $x : < \beta >$ que par un graphe avec renommages dont tous les sommets sont étiquetés par la même expression de renommage $< \beta > (x)$, où x peut être d'un type $< \gamma >$ différent de $< \beta >$.

Dans la proposition suivante, nous montrons dans quel cas un graphe avec renommages peut-être vu comme un graphe avec variable orbite et ainsi être instancié de manière équivalente selon l'instanciation des graphes avec variable orbite (voir définition 26 page 64) ou bien selon l'instanciation des graphes avec renommages (voir définition 39 page 81). À cette fin, nous allons utiliser les instanciations canoniques.

Proposition 40. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient $G = \{V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G}\}$ un graphe avec renommages sur la variable orbite $x : < \beta >$ et $O = \{V_O, E_O, s_O, t_O, l_{V_O}, l_{E_O}\}$ une orbite de type $< \beta >$.

L'instanciation canonique $I_{G,O} = \{V_I, E_I, s_I, t_I, l_{V_I}, l_{E_I}\}$ de G par O , définie comme suit, est une instanciation de G par O (voir définition 39 page 81) :

- $V_I = V_G \times V_O$;
- $E_I = E_{I,G} \uplus E_{I,O}$ où $E_{I,G} = E_G \times V_O$ est l'ensemble des arcs issus de G et $E_{I,O} \subset V_G \times E_O$ est l'ensemble des arcs issu de O qui contient l'ensemble des couples (v, d) tel que $\mu(l_{E_O}(d)) \neq _$ où $l_{V_G}(v) = < \gamma > (x)$ et μ est le morphisme qui définit la fonction de renommage $< \gamma > : < \beta > \rightarrow < \gamma >$;
- pour tout arc $(e, w) \in E_{I,G}$ issu de G :
 - $s_I((e, w)) = (s_G(e), w)$;
 - $t_I((e, w)) = (t_G(e), w)$;
 - $l_{E_I}((e, w)) = l_G(e)$;
- pour tout arc (v, d) issu de O (tel que $\mu(l_{E_O}(d)) \neq _$ où $l_{V_G}(v) = < \gamma > (x)$ et μ est le morphisme qui définit la fonction de renommage $< \gamma > : < \beta > \rightarrow < \gamma >$) :
 - $s_I((v, d)) = (v, s_O(d))$;
 - $t_I((v, d)) = (v, t_O(d))$;
 - $l_{E_I}((v, d)) = \mu(l_{E_O}(d))$.

Démonstration. Nous allons construire un isomorphisme de graphe $\varphi = (\varphi_V, \varphi_E)$ du graphe G^O vers le graphe $I_{G,O}$.

Construisons la fonction φ_V de l'ensemble V_{G^O} des sommets de G^O vers l'ensemble $V_I = V_G \times V_O$ des sommets de $I_{G,O}$ et montrons qu'il s'agit d'une bijection.

Tout d'abord, rappelons que la saturation et le passage au graphe anonyme préservent l'ensemble des sommets. Donc, G , G^{sat} et $(G^{sat})^{ano}$ ont le même ensemble de sommets, V_G . De même, O et $O^{sat}(\{\alpha\})$ ont le même ensemble de sommets, V_O . Et enfin, G^O et $(G^O)^{ano}$ ont le même ensemble de sommets, V_{G^O} .

De plus, d'après l'instanciation des graphes avec renommages (voir définition 39 page 81), tout sommet v de $(G^{sat})^{ano}$ a pour image par le morphisme f l'unique sommet de $D_n^{sat}(\{\alpha\})$. De même, tout sommet w de $O^{sat}(\{\alpha\})$ a pour image par le morphisme g l'unique sommet de $D_n^{sat}(\{\alpha\})$. Donc d'après la proposition 68 du produit fibré (voir l'annexe A.3.4), l'ensemble V_{G^O} des sommets de $(G^O)^{ano}$ est isomorphe au produit $V_G \times V_O$, des sommets de $(G^{sat})^{ano}$ et $O^{sat}(\{\alpha\})$.

Donc la fonction φ_V qui associe à tout sommet $u \in V_{G^O}$ de $(G^O)^{ano}$ le couple $(f'(u), g'(u)) \in V_G \times V_O$ de sommets de $O^{sat}(\{\alpha\})$ et $(G^{sat})^{ano}$ est bien une bijection.

Construisons la fonction φ_E de l'ensemble E_{G^O} des arcs de G^O vers l'ensemble E_I des arcs de $I_{G,O}$.

Tout d'abord, rappelons que le passage au graphe anonyme préserve l'ensemble des arcs et leurs accroches. Donc, G^{sat} et $(G^{sat})^{ano}$ ont le même ensemble d'arcs, $E_{G^{sat}}$, et les mêmes sources et cibles. Et, G^O et $(G^O)^{ano}$ ont le même ensemble d'arcs, E_{G^O} , et les mêmes sources et cibles.

De plus, la saturation ajoute des arcs, mais n'en supprime pas, et préserve les accroches et étiquettes. Donc l'ensemble E_O des arcs de O est inclus dans l'ensemble $E_{O^{sat}(\{\alpha\})}$ des arcs de $O^{sat}(\{\alpha\})$ et a les mêmes sources, cibles et étiquettes. Et l'ensemble des arcs E_G de G est inclus dans $E_{G^{sat}}$ l'ensemble des arcs de G^{sat} et $(G^{sat})^{ano}$ et a les mêmes sources, cibles et étiquettes.

Or, par définition du produit fibré, tout arc $c \in E_{G^O}$ de $(G^{sat})^{ano}$ a pour image par f' un arc d de $O^{sat}(\{\alpha\})$ et pour image par g' un arc e de $(G^{sat})^{ano}$.

Construisons φ_E comme suit :

- si e est un arc de $(G^{sat})^{ano}$ issu de G , $\varphi_E(c) = (e, s_{O^{sat}(\{\alpha\})}(d))$, qui est bien un arc de $E_{I,G}$;
- sinon e est un arc issu de la saturation, $\varphi_E(c) = (s_{(G^{sat})^{ano}}(e), d)$, qui est bien un arc de $E_{I,G}$, puisque aucun arc étiqueté par "_" n'est introduit par saturation.

Montrons que φ est un morphisme.

Soient $c \in E_{G^O}$ un arc de G^O (et/ou de $(G^O)^{ano}$), un arc d de $O^{sat}(\{\alpha\})$ image de c par f' et un arc e de $(G^{sat})^{ano}$ image de c par g' . Donc par définition du produit fibré,

$\varphi_V(s_{G^O}(c)) = (s_{G^{sat}}(e), s_{O^{sat}(\{\alpha\})}(d))$ et $\varphi_V(t_{G^O}(c)) = (s_{G^{sat}}(e), t_{O^{sat}(\{\alpha\})}(d))$. Et par définition de l'instanciation, $l_{G^O}(c) = l_{G^{sat}}(e)$.

Tout d'abord, supposons que e est un arc de G . Donc par définition de l'instanciation, l'image de e par f est l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par α . Donc par définition du produit fibré, d a aussi pour image par g l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par α . Donc par définition de l'instanciation, d est une boucle issue de la saturation. Donc $s_{O^{sat}(\{\alpha\})}(d) = t_{O^{sat}(\{\alpha\})}(d)$. Notons w ce sommet de O . Donc $s_I(\varphi_E(c)) = \varphi_V(s_{G^O}(c)) = (s_G(e), w)$, $t_I(\varphi_E(c)) = \varphi_V(t_{G^O}(c)) = (t_g(e), w)$ et $l_{E_I}(\varphi_E(c)) = \varphi_V(l_{E_{G^O}}(c)) = l_G(e)$.

Supposons maintenant, que e est la boucle $e_{v,k}$ issue de la saturation (où v est le sommet de G^{sat} source et origine de $e = e_{v,k}$ et $k \in [1, |\beta|]$ un indice). Donc $s_{G^{sat}}(e) = t_{G^{sat}}(e)$. Notons γ l'étiquette de v dans G . Par définition de la saturation, l'étiquette de $e = e_{v,k}$ dans G^{sat} est $\gamma(k)$ qui est différente de "_". Par définition de l'instanciation, l'image de $e = e_{v,k}$ par f est l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par $\beta(k)$. Donc par définition du produit fibré, d a pour image par g l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par $\beta(k)$. Donc par définition de l'instanciation, d est étiqueté par $\beta(k)$. Donc l'étiquette $\gamma(k)$ de e (et donc de c) est bien le renommage de l'étiquette $\beta(k)$ de d . Donc $s_{G^O}(\varphi_E(c)) = \varphi_V(s_{G^O}(c)) = (v, s_O(d))$, $t_{G^O}(\varphi_E(c)) = \varphi_V(t_{G^O}(c)) = (v, t_O(d))$ et $l_{E_{G^O}}(\varphi_E(c)) = \varphi_V(l_{E_{G^O}}(c)) = \gamma(k)$.

φ est donc bien un morphisme.

Montrons que φ_E est une bijection.

Soit (e, w) un arc issu de G de $I_{G,O}$. e est donc un arc de G . Donc comme nous l'avons vu ci-dessus, l'image de e par f est l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par α . Or par définition de la saturation, le sommet w de $O^{sat}(\{\alpha\})$ a une unique boucle $e_{w,1}$ étiqueté par α . Et par définition de l'instanciation, $e_{w,1}$ est le seul arc incident à v dont l'image par g est l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par α . Donc, d'après la proposition 68 du produit fibré (voir l'annexe A.3.4), il existe un unique arc c de $(G^O)^{ano}$, dont l'image par f' est $e_{w,1}$ et l'image par g' est e .

Soit (v, d) un arc issu de O de $I_{G,O}$. Notons $k \in [1, |\beta|]$ l'indice tel que $\beta(k)$ est l'étiquette de d dans O . Par définition de l'instanciation, l'image de d par g est l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par $\beta(k)$. Soit γ l'étiquette de v dans G . Par hypothèse, $\gamma(k) \neq _$. Donc, par définition de la saturation, $e_{v,k}$ est une boucle issue de la saturation de G^{sat} . Or β est type d'orbite, donc toutes les lettres de β sont différentes. Et donc, par définition de l'instanciation, $e_{v,k}$ le seul arc de $(G^{sat})^{ano}$ incident à v dont l'image par f est l'unique arc de $D_n^{sat}(\{\alpha\})$ étiqueté par $\beta(k)$. Donc, d'après la proposition 68 du produit fibré (voir l'annexe A.3.4), il existe un unique arc c de $(G^O)^{ano}$, dont l'image par f' est d et l'image par g' est $e_{v,k}$.

Donc φ_E est une bijection.

Donc φ est un isomorphisme. Et donc G^O est isomorphe à $I_{G,O}$. □

Nous pouvons à présent montrer l'équivalence des graphes avec renommages et des graphes avec variable orbite.

Proposition 41. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G un graphe avec renommages sur la variable orbite $x : \langle \beta \rangle$ tel que :

- tous les sommets de G sont étiquetés par la même expression $\langle \beta \rangle (x)$;
- tous les arcs de G sont étiquetés par des étiquettes de $\mathcal{A}_n \setminus \Sigma_\beta$.

et O une orbite de type $\langle \beta \rangle$.

Notons G' le graphe avec variable orbite qui a la même réalisation graphique que G . Par définition, G' a les mêmes sommets, arcs, sources, cibles et étiquettes d'arcs que G et a tous ses sommets étiquetés par $x : \langle \beta \rangle$.

Les instantiations G^O de G selon la définition 39 – c'est-à-dire dans le cas des graphes avec renommages – et G'^O de G' selon la définition 26 – c'est-à-dire dans le cas des graphes avec variable orbite – sont identiques à l'isomorphisme près.

Démonstration. Montrons que les instantiations canoniques $I_{G,O}$ et $I_{G',O}$ sont identiques.

Tout d'abord, G et G' ont le même ensemble de sommets ; nous le notons V_G . Donc $I_{G,O}$ et $I_{G',O}$ ont le même ensemble de sommets $V_G \times V_O$. De la même manière, G et G' ont le même ensemble d'arcs ; nous le notons E_G . Donc $I_{G,O}$ et $I_{G',O}$ ont le même ensemble d'arcs issus respectivement de G et de G' , à savoir $E_G \times E_O$. De plus, G a tous ses sommets étiquetés par $\langle \beta \rangle (x)$ et β ne contient aucune occurrence de « $_$ ». Donc l'ensemble des arcs de $I_{G,O}$ issus de O est $V_G \times E_O$, qui est aussi l'ensemble des arcs de $I_{G',O}$.

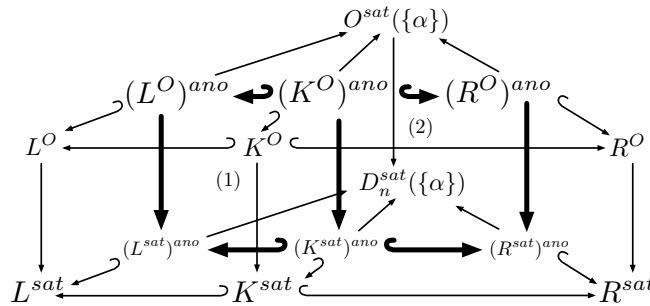
Par conséquent, les instantiations canoniques $I_{G,O}$ et $I_{G',O}$ sont identiques et donc toutes les instantiations G^O et G'^O sont isomorphes. □

3.3.5.4 Instanciation des règles de transformation avec renommages

Intuitivement, l'instanciation d'une production avec renommages $p : L \leftarrow K \rightarrow R$ par une orbite O est une production $p^O : L^O \leftarrow K^O \rightarrow R^O$ où L^O , K^O et R^O sont respectivement les instanciations de la partie gauche L , de l'interface K et de la partie droite R par l'orbite O . Comme précédemment, les instanciations L^O , K^O et R^O doivent être construites avec la même convention de nommage des sommets et des arcs, afin de reconstruire les inclusions de p^O .

Définition 42 (Instanciation d'une production avec renommages). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient $p : L \leftarrow K \rightarrow R$ une production avec renommages sur la variable orbite $x : \langle \beta \rangle$ et O une orbite de type $\langle \beta \rangle$.

L'instanciation de p par l'orbite O est la production $p^O : L^O \leftarrow K^O \rightarrow R^O$ dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\emptyset, \mathcal{A}_n)$ telle que les diagrammes (1) et (2) – représentés en gras sur la figure ci-dessous – commutent :



Les inclusions de $(K^O)^{ano}$ dans $(L^O)^{ano}$ et $(R^O)^{ano}$ sont garanties par des arguments analogues au cas de l'instanciation d'une production avec variable orbite (voir définition 29 page 66). Les inclusions $K \hookrightarrow L$ et $K \hookrightarrow R$ entraînent les inclusions $K^{sat} \hookrightarrow L^{sat}$ et $K^{sat} \hookrightarrow R^{sat}$. En effet, pour tout sommet u de K , si u est étiqueté par $\langle \gamma \rangle (x)$ dans K , il l'est également dans L et R et est donc saturé par γ dans les trois graphes L , K et R . Si au contraire, u n'est pas étiqueté dans K , aucune boucle incidente à u n'est ajoutée par saturation dans K^{sat} . Notons que ces inclusions sont bien sûr préservées par les graphes anonymes, $(K^{sat})^{ano} \hookrightarrow (L^{sat})^{ano}$ et $(K^{sat})^{ano} \hookrightarrow (R^{sat})^{ano}$.

La commutativité des diagrammes (1) et (2) permet de garantir les inclusions $(K^O)^{ano} \hookrightarrow (L^O)^{ano}$ et $(K^O)^{ano} \hookrightarrow (R^O)^{ano}$. Ces inclusions sont préservées par l'ajout des étiquettes. En effet, comme nous l'avons déjà vu, K^{sat} est inclus dans L^{sat} et R^{sat} . Ainsi, K^O est inclus dans L^O et R^O . De fait, $p^O : L^O \leftarrow K^O \rightarrow R^O$ est bien une production.

L'instanciation p_{cone}^O de la règle de 3-cône p_{cone} (voir Fig. 3.30(a)) par l'orbite O de la Fig. 3.30(b) – qui représente le coin d'une face – est illustrée en Fig. 3.30(c). Cette règle est analogue à la règle de cône sur un coin de face, déjà présentée en Fig. 3.20(a) page 69.

3.3.5.5 Chanfreinage topologique

Dans le chapitre 1, nous avons introduit l'opération de chanfreinage topologique, qui permet d'adoucir les bords des objets (voir section 1.5.2). Nous avons illustré cette opération des cartes généralisées dans le cas de l'arrondi d'un sommet de volume. Nous traduisons le chanfreinage d'un sommet de volume en la production p_{chan} illustrée en Fig. 3.31(a). L'instanciation $p_{chan}^{O_s}$ de p_{chan} par l'orbite O_s de type $\langle \alpha_1 \alpha_2 \rangle$ (voir Fig. 3.31(b)) est illustrée en Fig. 3.31(c). Dans l'instanciation, on remarque que l'orbite O_s filtrée par le sommet 1 de L_{chan} est de type $\langle \alpha_1 \alpha_2 \rangle$. Dans le membre droite, l'orbite associée au sommet 1 de R_{chan} est l'orbite renommée $\langle _ \alpha_2 \rangle (O_s)$. Le renommage de l'orbite associée au

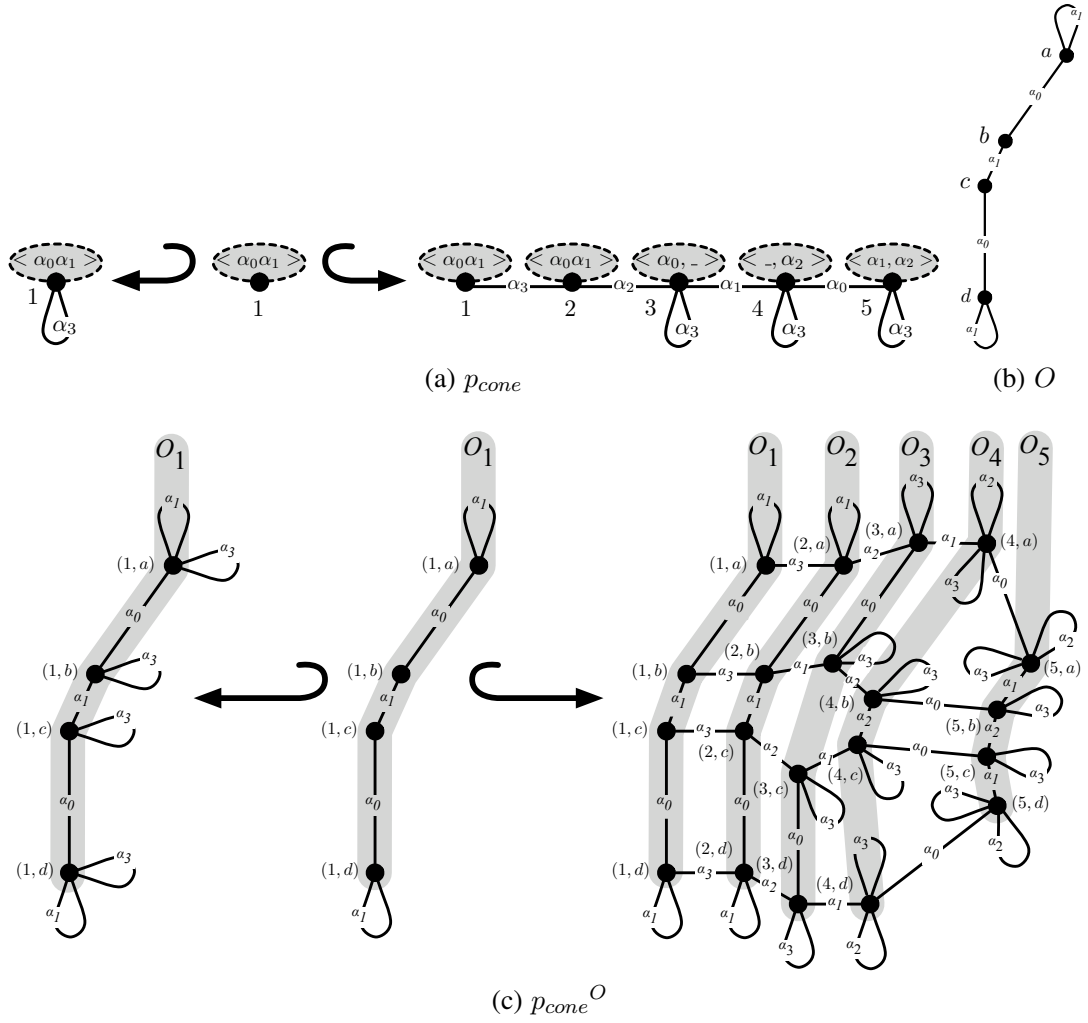


FIGURE 3.30 – Instantiation p_{cone}^O de la règle p_{cone} par une orbite O

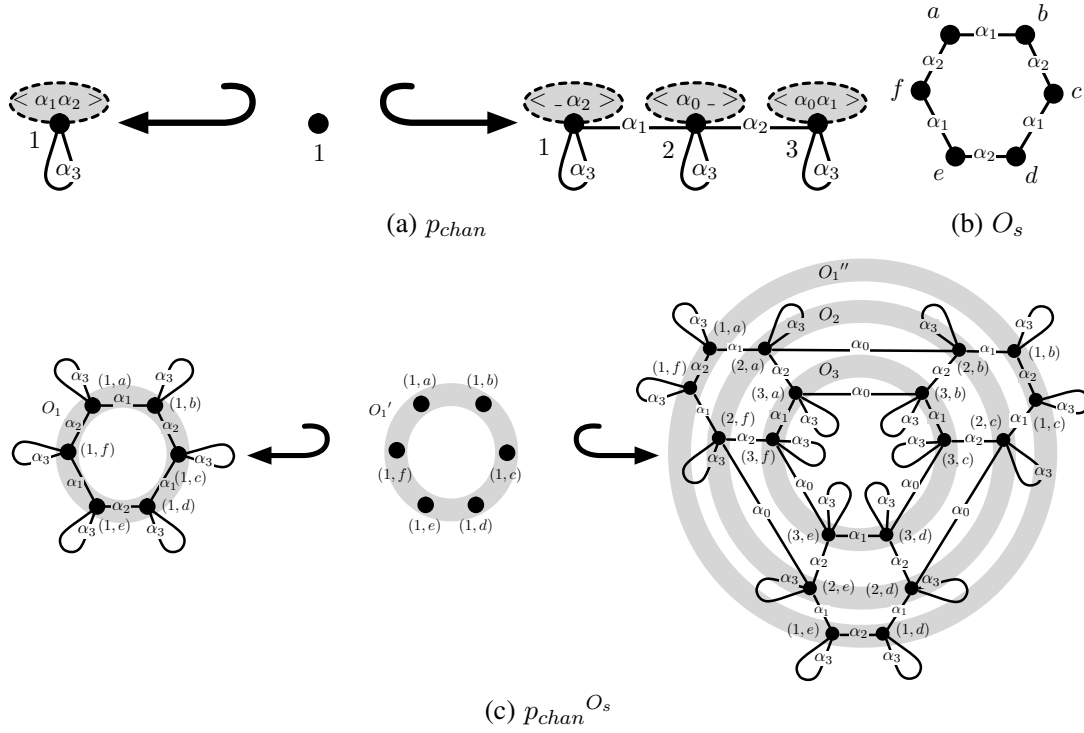
sommet 1 de l'interface K_{chan} , nous conduit à ne pas étiqueter ce sommet, ce sans quoi p_{chan} ne serait pas une règle de transformation. En effet, une étiquette de A ne pourrait être conservée à la fois dans L_{chan} et R_{chan} .

3.4 Renommage d'orbites et cohérence des cartes généralisées

Dans la section 2.3.4, nous avons étendu les contraintes d'arcs incidents, de non orientation et de cycles des cartes généralisées aux règles de transformation de graphes. Nous savons donc écrire des règles de transformation qui préservent les contraintes de cohérence des cartes généralisées. Dans cette section, nous étendons ces contraintes aux règles avec renommages d'orbites.

3.4.1 Contrainte d'arcs incidents

Dans un premier temps, nous étendons la contrainte d'arcs incidents aux graphes avec renommages. Lors de l'instanciation d'un graphe avec renommages, les sommets sont dupliqués en des sommets munis d'arcs incidents étiquetés par des étiquettes issues du graphe et des renommages. Ainsi, la contrainte d'arcs incidents des graphes avec renommages est une extension de la contrainte d'arcs incidents sur les graphes partiellement étiquetés où l'on considère simultanément les étiquettes des arcs et celles des

FIGURE 3.31 – Instantiation $p_{chan}^{O_s}$ de la règle p_{chan} par une orbite O_s

renommages.

Définition 43 (contrainte d'arcs incidents sur les graphes avec renommages). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soit G un graphe avec renommages sur la variable $x : \langle \beta \rangle$ et $\mathcal{A}_n' \subset \mathcal{A}_n$ un ensemble d'étiquettes d'arcs.

Soit v un sommet de G . Le sommet v satisfait la contrainte d'arcs incidents $\text{Incede}_{v,G}(\mathcal{A}_n')$ si et seulement si, pour toute étiquette $l \in \mathcal{A}_n'$ une et une seule de ces deux propriétés est vérifiée :

- v a un unique arc incident étiqueté par l ;
- v est étiqueté par une fonction de renommage $\langle \gamma \rangle$ tel que le mot γ possède une unique occurrence de l ;

et pour toute étiquette l' de $\mathcal{A}_n \setminus \mathcal{A}_n'$, v ne possède pas d'arc incident étiqueté par l' et γ n'a pas d'occurrence de l' .

Le graphe G satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n')$ si et seulement si pour tout sommet v de G , $\text{Incede}_{v,G}(\mathcal{A}_n')$ est satisfaite.

Soit H un graphe avec renommages sur la variable $x : \langle \beta \rangle$. Nous étendons naturellement les contraintes d'arcs incidents aux structures issues de la soustraction $G \setminus H$. Notons $\text{Incede}_{v,G \setminus H}(\mathcal{A}_n')$ la contrainte d'arcs incidents pour un sommet v de $G \setminus H$ sur l'ensemble \mathcal{A}_n' . La soustraction de H à G , $G \setminus H$, satisfait la contrainte d'arcs incidents $\text{Incede}_{G \setminus H}(\mathcal{A}_n')$ si et seulement si pour tout sommet v de $G \setminus H$, $\text{Incede}_{v,G \setminus H}(\mathcal{A}_n')$ est satisfaite.

Prenons l'exemple de la Fig. 3.32 avec $\mathcal{A}_3' = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$. Le graphe G de la Fig. 3.32(a) satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_3')$. En effet, les étiquettes α_0 et α_1 apparaissent dans le renommage qui étiquette l'unique sommet u de G mais pas parmi les arcs incidents à u . De plus, les étiquettes α_2 et α_3 étiquettent les deux boucles incidentes à u sans apparaître dans le renommage. Le sommet v du graphe G' de la Fig. 3.32(b) ne possède pas d'arc incident étiqueté par α_3 et est étiqueté

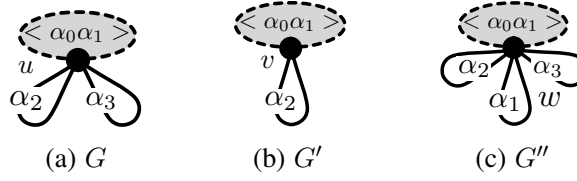


FIGURE 3.32 – contrainte d’arcs incidents sur les graphes avec renommages

par un renommage qui ne contient pas non plus α_3 . Ainsi, G' ne satisfait pas la contrainte d’arcs incidents $\text{Incede}_{G'}(\mathcal{A}_3')$. Enfin, le graphe G'' ne satisfait pas la contrainte d’arcs incidents $\text{Incede}_{G''}(\mathcal{A}_3')$ car l’étiquette α_1 apparaît à la fois dans le renommage qui étiquette le sommet w et parmi ses boucles incidentes.

Démontrons à l’aide de la proposition suivante, que l’instanciation des graphes avec renommages préserve la contrainte d’arcs incidents.

Proposition 44. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G un graphe avec renommages sur la variable $x : \langle \beta \rangle$, et O une orbite de type $\langle \beta \rangle$.

Soit G^O , l’instanciation de G par O , définie par le diagramme suivant :

$$\begin{array}{ccccc}
 G^O & \xleftarrow{h'} & H & \xrightarrow{f'} & O^{\text{sat}}(\{\alpha\}) \\
 \downarrow g'' & (1) & \downarrow g' & (2) & \downarrow g \\
 G^{\text{sat}} & \xleftarrow{h} & (G^{\text{sat}})^{\text{ano}} & \xrightarrow{f} & (D_n)^{\text{sat}}(\{\alpha\})
 \end{array}$$

Pour tout sommet u de G^O , s’il existe un ensemble d’étiquettes d’arcs \mathcal{A}_n' tel que la contrainte d’arcs incidents $\text{Incede}_{g''(u),G}(\mathcal{A}_n')$ est satisfaite alors $\text{Incede}_{u,G^O}(\mathcal{A}_n')$ est satisfaite³.

Démonstration. Montrons que l’instanciation canonique $I_{G,O}$ vérifie la proposition.

Soit $u = (v, w)$ un sommet de $I_{G,O}$, où v est un sommet de G et w un sommet de O . Notons $\langle \gamma \rangle (x)$ l’étiquette de v dans G , et μ le morphisme qui définit le renommage $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$. Par construction, $g''(u) = v$. Supposons qu’il existe un ensemble d’étiquettes d’arcs \mathcal{A}_n' tel que la contrainte d’arcs incidents $\text{Incede}_{v,G}(\mathcal{A}_n')$ est satisfaite. Montrons que $\text{Incede}_{(v,w),I_{G,O}}(\mathcal{A}_n')$ est satisfaite.

Par construction, les arcs de $I_{G,O}$ de source (v, w) sont de deux formes :

- l’arc (v, d) étiqueté par $\mu(l')$, où d est un arc de O de source v et d’étiquette l' , avec $\mu(l') \neq _$;
- l’arc (e, w) étiqueté par l , où e est un arc de G de source v et d’étiquette l .

Or O est une orbite. Donc la contrainte d’arcs incidents $\text{Incede}_{w,O}(\Sigma_\beta)$ est satisfaite. Donc pour toute étiquette $l' \in \Sigma_\beta$, il existe un unique indice $k \in [1, |\beta|]$ tel que $\beta(k) = l'$. Or, comme par hypothèse $\text{Incede}_{v,G}(\mathcal{A}_n')$ est satisfaite, pour toute étiquette $l \in \Sigma_\gamma \setminus \{ _ \}$, il existe un unique indice $k \in [1, |\beta|]$ tel que $\gamma(k) = l$ et $\mu(\beta(k)) = l$. Donc tous les arcs de $I_{G,O}$ de source (v, w) et de la forme (v, d) ont des étiquettes de Σ_γ toutes différentes et réciproquement, pour toute étiquette l de $\Sigma_\gamma \setminus \{ _ \}$, il existe un unique arc d de O incident à w et étiqueté par l’unique étiquette $l' \in \Sigma_\beta$ tel que $\mu(l') = l$.

De plus, comme par hypothèse $\text{Incede}_{v,G}(\mathcal{A}_n')$ est satisfaite, et donc pour toute étiquette

3. Rappelons que les ensemble de sommets de G et de G^{sat} sont les mêmes.

$l \in \mathcal{A}_n' \setminus \Sigma_\gamma$, il existe un unique arc e de G de source v et d'étiquette l . Et G n'a pas d'autres arcs incidents à v . Donc (e, w) est l'unique arc de $I_{G,O}$ qui a pour source (v, w) et pour étiquette l . Et tous les arcs de cette deuxième forme sont étiquetés par des étiquettes de $\mathcal{A}_n' \setminus \Sigma_\gamma$.

Donc la contrainte d'arcs incidents $\text{Incede}_{(v,w), I_{G,O}}(\mathcal{A}_n')$ est satisfaite.

Or G^O est isomorphe à $I_{G,O}$ (voir la proposition 40). Donc G^O vérifie aussi la proposition. \square

Théorème 45. Soient $n \geq 0$ une dimension et G un graphe topologique de dimension n . Soient β un mot de \mathcal{A}_n^* , $p : L \hookrightarrow K \hookrightarrow R$ une production avec renommages sur la variable $x : \langle \beta \rangle$ et O une orbite de type $\langle \beta \rangle$.

Si G satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$ et p satisfait la contrainte d'arcs incidents $\text{Incede}_p(\mathcal{A}_n)$ suivante, alors tout graphe topologique H résultat de la transformation directe $G \xrightarrow{p^O, m} H$ pour un morphisme de filtrage m , satisfait la contrainte d'arcs incidents $\text{Incede}_H(\mathcal{A}_n)$.

$\text{Incede}_p(\mathcal{A}_n)$: les contraintes d'arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$ et $\text{Incede}_{R \setminus K}(\mathcal{A}_n)$ sont satisfaites et pour chaque sommet v de K , il existe un ensemble d'étiquettes d'arcs $\mathcal{A}_n' \subset \mathcal{A}_n$ tel que les contraintes d'arcs incidents $\text{Incede}_{v,L}(\mathcal{A}_n')$ et $\text{Incede}_{v,R}(\mathcal{A}_n')$ sont satisfaites.

Démonstration. Supposons que G satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$ et que p satisfait la contrainte d'arcs incidents $\text{Incede}_p(\mathcal{A}_n)$.

Montrons que la contrainte d'arcs incidents $\text{Incede}_{pO}(\mathcal{A}_n)$ est satisfaite.

Soit v un sommet de $L \setminus K$. Par hypothèse, la contrainte d'arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$ est satisfaite. Donc par définition, la contrainte $\text{Incede}_{v,L \setminus K}(\mathcal{A}_n)$ est satisfaite. Or, comme v n'est pas un sommet de K , aucun arc de K n'est incident à v . Donc l'ensemble des arcs incidents à v dans L et dans $L \setminus K$ sont les mêmes. Donc la contrainte $\text{Incede}_{v,L}(\mathcal{A}_n)$ est satisfaite. Donc d'après la proposition 44, la contrainte $\text{Incede}_{u,L^O}(\mathcal{A}_n)$ est satisfaite, pour tout sommet u de L^O isomorphe à un sommet (v, w) de l'instanciation canonique $I_{L,O}$ (où w est un sommet de O). Or, comme v n'est pas un sommet de K , (v, w) n'est pas non plus un sommet de l'instanciation canonique $I_{K,O}$ (voir la proposition 40). Et donc par isomorphisme, u n'est pas un sommet de K^O . Donc, pour les mêmes raisons que précédemment, les arcs de L^O incidents à u sont les mêmes que les arcs de $L^O \setminus K^O$ incidents à u . Donc la contrainte $\text{Incede}_{u,L^O \setminus K^O}(\mathcal{A}_n)$ est satisfaite pour tout sommet u de $L^O \setminus K^O$. Donc, la contrainte d'arcs incidents $\text{Incede}_{L^O \setminus K^O}(\mathcal{A}_n)$ est satisfaite.

Par un raisonnement identique, on peut montrer que la contrainte d'arcs incidents $\text{Incede}_{R^O \setminus K^O}(\mathcal{A}_n)$ est satisfaite.

Soit v un sommet de K . Par hypothèse, il existe un ensemble d'étiquettes d'arcs \mathcal{A}_n' tel que les contraintes d'arcs incidents $\text{Incede}_{v,L}(\mathcal{A}_n')$ et $\text{Incede}_{v,R}(\mathcal{A}_n')$ sont satisfaites. Soit w un sommet de O . Notons u le sommet de K^O isomorphe au sommet (v, w) de l'instanciation canonique $I_{K,O}$ (voir la proposition 40). D'après la proposition 44, les contraintes $\text{Incede}_{u,L^O}(\mathcal{A}_n')$ et $\text{Incede}_{u,R^O}(\mathcal{A}_n')$ sont satisfaites.

Donc la contrainte d'arcs incidents $\text{Incede}_{pO}(\mathcal{A}_n)$ est satisfaite.

Donc, d'après la proposition 16, page 41, le graphe topologique H , résultat de la transformation directe $G \xrightarrow{p^O, m} H$, satisfait la contrainte d'arcs incidents $\text{Incede}_H(\mathcal{A}_n)$. \square

Illustrons la proposition précédente à l'aide de l'opération de cône. Rappelons que la règle de cône p_{cone} ainsi que son instanciation $p_{\text{cone}}^{O_{\text{cone}}}$ par l'orbite O_{cone} de type $\langle \alpha_0 \alpha_1 \rangle$ sont illustrées en Fig. 3.30 page 88. Remarquons tout d'abord qu'aucun sommet n'est supprimé lors de l'application de la règle de cône à un graphe, ainsi $\text{Incede}_{L_{\text{cone}} \setminus K_{\text{cone}}}(\mathcal{A}_3)$ est satisfaite par définition. Les sommets ajoutés

lors de l'application de la règle de cône sont issus des sommets 2 à 5 de $R_{cone} \setminus K_{cone}$. Or, chacun de ces sommets vérifie la contrainte d'arc incidents. Par exemple, le sommet 3 possède exactement trois arcs incidents étiquetés par α_1 , α_2 et α_3 . De plus, le renommage $\langle \alpha_0 _ \rangle$, qui étiquette le sommet 3, présente une unique occurrence de α_0 . De la même manière, le sommet 2 possède exactement deux arcs incidents étiquetés par α_2 et α_3 . Et de plus, le renommage $\langle \alpha_0 \alpha_1 \rangle$ du sommet 2 possède une unique occurrence des étiquettes α_0 et α_1 . La contrainte d'arcs incidents $Incide_{R_{cone} \setminus K_{cone}}(\mathcal{A}_3)$ est donc satisfaite.

Intéressons-nous à présent à l'unique sommet 1 de K_{cone} . D'après la contrainte d'arcs incidents des productions avec renommages d'orbites introduite dans la proposition précédente, il doit exister un ensemble d'étiquette d'arcs \mathcal{A}_3' tel que $Incide_{1, L_{cone}}(\mathcal{A}_3')$ et $Incide_{1, R_{cone}}(\mathcal{A}_3')$ soit satisfaite. Dans notre cas, nous pouvons prendre $\mathcal{A}_3' = \{\alpha_0, \alpha_1, \alpha_2\}$. En effet, dans L_{cone} et R_{cone} , le sommet 1 possède un unique arc incidents étiqueté par α_3 et est étiqueté par un renommage $\langle \alpha_0 \alpha_1 \rangle$. Ainsi, la règle de cône satisfait la contrainte d'arcs incidents $Incide_{p_{cone}}(\mathcal{A}_3)$.

D'après la démonstration de la proposition précédente, l'instanciation $p_{cone}^{O_{cone}}$ de p_{cone} par l'orbite O_{cone} satisfait également la contrainte $Incide_{p_{cone}^{O_{cone}}}(\mathcal{A}_3)$ introduite dans la proposition 16 page 41. Remarquons notamment que tous les sommets ajoutés de $R_{cone}^{O_{cone}} \setminus K_{cone}^{O_{cone}}$ – c'est-à-dire ceux issus dont les noms sont construits sur un couple (v, w) avec $v \in V_{R_{cone}} \setminus \{1\}$ et $w \in V_{O_{cone}}$ – satisfont la contrainte d'arc incidents. Par exemple, le sommet $(4, a)$ possède un arc incident étiqueté par α_1 – respectivement α_0 – issu de l'arc de même étiquette qui connecte le sommet 4 au sommet 3 – respectivement le sommet 4 au sommet 5 – dans R_{cone} . Il possède également un arc incident étiqueté par α_3 issu de la boucle α_3 incidente au sommet 3. Enfin, le sommet $(3, a)$ possède une boucle incidente étiquetée par α_2 issue du renommage de la boucle α_1 incidente au sommet a dans O_{cone} . Ainsi, $Incide_{R_{cone}^{O_{cone}} \setminus K_{cone}^{O_{cone}}}(\mathcal{A}_3)$ est satisfaite. De plus, pour tout sommet conservé $(1, w)$ de $K_{cone}^{O_{cone}}$, il existe un ensemble d'étiquettes \mathcal{A}_3' tel que les contraintes d'arcs incidents de sommet $Incide_{(1, w), L_{cone}^{O_{cone}}}(\mathcal{A}_3')$ et $Incide_{(1, w), R_{cone}^{O_{cone}}}(\mathcal{A}_3')$ sont satisfaites. Par exemple, dans $L_{cone}^{O_{cone}}$ et $R_{cone}^{O_{cone}}$, le sommet $(1, a)$ possède exactement trois arcs incidents étiquetés par α_0 à α_3 . L'arc étiqueté par α_3 est issu de G tandis que les deux arcs étiquetés par α_0 et α_1 sont issus du renommage identité des arcs de O_{cone} .

La proposition 16 nous indique donc que pour tout graphe topologique G de dimension n , le graphe topologique H résultat de la transformation directe $G \xrightarrow{p_{cone}^{O_{cone}}, m} H$ pour un morphisme de filtrage m , satisfait la contrainte d'arcs incidents $Incide_H(\mathcal{A}_3)$.

Nous pouvons illustrer la proposition précédente de manière analogue en nous appuyant sur la règle de chanfreinage p_{chan} (voir Fig. 3.31 page 89). En effet, celle-ci satisfait également la contrainte d'arcs incidents des règles avec renommages d'orbites. La principale différence qui distingue p_{cone} de p_{chan} est le renommage – dans p_{chan} – de l'orbite filtrée dans la partie gauche de la règle. En effet, l'orbite filtrée de type $\langle \alpha_1 \alpha_2 \rangle$ est renommée en un graphe de type $\langle _ \alpha_2 \rangle$. Dans L_{chan} , c'est le sommet 1 muni de son renommage $\langle \alpha_1 \alpha_2 \rangle$ qui permet de filtrer cette orbite. Le sommet 1, qui est un sommet conservé de K_{chan} , est étiqueté par le renommage $\langle _ \alpha_2 \rangle$ dans R_{chan} . Remarquons qu'en prenant $\mathcal{A}_3' = \{\alpha_1, \alpha_2, \alpha_3\}$, les contraintes $Incide_{1, L_{chan}}(\mathcal{A}_3')$ et $Incide_{1, R_{chan}}(\mathcal{A}_3')$ sont bien satisfaites. Par conséquent, dans l'instanciation $p_{chan}^{O_{chan}}$ de p_{chan} par l'orbite O_{chan} , on remarque que pour tout sommet $(1, w)$ – avec $w \in V_{O_{chan}}$ – issu de la duplication du sommet 1 le long de O_{chan} , les contraintes $Incide_{(1, w), L_{chan}^{O_{chan}}}(\mathcal{A}_3')$ et $Incide_{(1, w), R_{chan}^{O_{chan}}}(\mathcal{A}_3')$ sont également satisfaites.

3.4.2 Préservation des pré-G-cartes

Nous venons de voir que la condition d'arcs incidentes des règles de transformation avec renommages préserve la contrainte d'arcs incidents des graphes avec renommages par transformation directe.

Pour garantir la préservation des pré-G-cartes, il nous reste à étudier la préservation de la non orientation des graphes topologiques.

Proposition 46. *Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G un graphe non orienté avec renommages sur la variable $x : \langle \beta \rangle$, et O une orbite de type $\langle \beta \rangle$.*

L'instanciation de G^O est non orientée.

Démonstration. Soit c un arc de G^O . Notons u et u' les sommets de G^O respectivement source et cible de c . Notons l l'étiquette de d dans G^O . Montrons que G^O possède un arc c' inverse de c , c'est-à-dire de source u' , de cible u et d'étiquette l .

Supposons tout d'abord que c est un arc issu de G .

C'est-à-dire que c est isomorphe à un arc (e, w) de l'instanciation canonique $I_{G,O}$ (voir la proposition 40), où e est un arc de G et w un sommet de O . Notons v et v' respectivement les sommets source et cible de e dans G . D'après la proposition 40, u et u' sont respectivement isomorphes aux sommets (v, w) et (v', w) de l'instanciation canonique $I_{G,O}$. Et e est étiqueté par l dans G .

Or G est non orienté. Donc G a un arc e' inverse de e de source v' , de cible v et d'étiquette l . Donc, d'après la proposition 40, l'instanciation canonique $I_{G,O}$ a un arc (e', w) de source (v', w) , de cible (v, w) et d'étiquette l . L'arc (e', w) de $I_{G,O}$ est donc bien inverse de (e, w) .

Donc c' est l'arc isomorphe à (e', w) .

Supposons maintenant, que c est un arc issu de O .

C'est-à-dire que c est isomorphe à un arc (v, d) de l'instanciation canonique $I_{G,O}$, où v est un sommet de G et d un arc de O . Notons $\langle \gamma \rangle (x)$ l'étiquette de v dans G et μ le morphisme de mots qui définit la fonction de renommage $\langle \gamma \rangle : \langle \beta \rangle \rightarrow \langle \gamma \rangle$. Notons w et w' respectivement l'arc source et l'arc cible de d dans O . De plus, par construction, si l' est l'étiquette de e dans O , alors $l = \mu(l')$.

Or, comme toutes les orbites, O est non orientée. Donc O a un arc d' inverse de d , c'est-à-dire de source w' , de cible w et d'étiquette l' . Donc, d'après la proposition 40, l'instanciation canonique $I_{G,O}$ a un arc (v, d') de source (v, w') , de cible (v, w) et d'étiquette $l = \mu(l')$. (v, d') est donc bien un arc inverse de (v, d) .

Donc c' est l'arc isomorphe à (v, d') .

□

Proposition 47. *Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G un graphe topologique de dimension n , $p : L \leftrightarrow K \hookrightarrow R$ une production avec renommages sur la variable $x : \langle \beta \rangle$ et O une orbite de type $\langle \beta \rangle$.*

Si G est un graphe non orienté et p est une règle non orientée (c'est-à-dire si L , K et R sont des graphes non orientés), alors tout graphe topologique H résultat de la transformation directe $G \Rightarrow_{p^O, m} H$ pour un morphisme de filtrage m , est un graphe non orienté.

Démonstration. Supposons que G et p soient non orientés. D'après la proposition 46, L^O , K^O et R^O sont non orientés, donc p^O est non orientée. Donc d'après la proposition 17, H résultat de la transformation directe $G \Rightarrow_{p^O, m} H$ est un graphe non orienté.

□

Sur l'exemple du cône, voir Fig. 3.30 page 88, on remarque qu'une production non orientée avec renommages s'instancie en une production non orientée. Par conséquent, lorsque nous l'appliquons à un graphe non orienté, nous obtenons un graphe non orienté.

Corollaire 48. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G une pré- G -carte de dimension n , $p : L \hookleftarrow K \hookrightarrow R$ une production avec renommages sur la variable $x : \langle \beta \rangle$ et O une orbite de type $\langle \beta \rangle$.

Si p est une règle non orientée qui satisfait la contrainte d'arcs incidents $\text{Incede}_p(\mathcal{A}_n)$, alors tout graphe topologique H résultat de la transformation directe $G \Rightarrow_{p^O, m} H$ pour un morphisme de filtrage m , est une pré- G -carte.

Démonstration. Le corollaire est la conséquence directe de la préservation de la contrainte d'arcs incidents (voir le théorème 45) et de la préservation de la non orientation (voir la proposition 47 précédente). \square

3.4.3 Condition d'arcs pendants

Nous avons vu dans le chapitre 2 que l'existence d'une transformation directe n'est garantie que sous la condition d'arcs pendants (voir section 2.2.2 page 35). Or, nous savons que les contraintes de cohérence des cartes généralisées permettent de garantir statiquement la satisfaction de cette condition et donc l'applicabilité de nos règles de transformation (voir théorème 15 page 40). Nous devons étendre ce théorème au cas des règles avec renommages.

Théorème 49. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G un graphe topologique de dimension n qui soit non orienté et qui satisfait la contrainte d'arcs incidents $\text{Incede}_G(\mathcal{A}_n)$, $p : L \hookleftarrow K \hookrightarrow R$ une production avec renommages sur la variable $x : \langle \beta \rangle$ telle que L et K soient non orientés, et O une orbite de type $\langle \beta \rangle$ telle qu'il existe un morphisme de filtrage $m : L^O \rightarrow G$.

Le morphisme de filtrage m satisfait la condition d'arcs pendants si $L \setminus K$ satisfait la contrainte d'arcs incidents $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$.

Démonstration. Supposons que $L \setminus K$ satisfait la contrainte $\text{Incede}_{L \setminus K}(\mathcal{A}_n)$ et montrons que le morphisme de filtrage m satisfait la condition d'arcs pendants.

Par hypothèse, L et K sont non orientés. Donc, d'après la proposition 46, L^O et K^O sont non orientés.

Par définition, pour tout sommet v de $L \setminus K$, la contrainte $\text{Incede}_{v, L \setminus K}(\mathcal{A}_n)$ est satisfaite. Donc $\text{Incede}_{v, L}(\mathcal{A}_n)$ est satisfaite, car tout arc de source v est un arc de $L \setminus K$.

Or, d'après la proposition 44, pour tout sommet u de $L^O \setminus K^O$, $\text{Incede}_{u, L^O}(\mathcal{A}_n)$ est satisfaite. Donc $\text{Incede}_{u, L^O \setminus K^O}(\mathcal{A}_n)$ est satisfaite, car tout arc de source u est un arc de $L^O \setminus K^O$.

Donc, d'après le théorème 15 page 40, le morphisme de filtrage satisfait la condition d'arcs pendants. \square

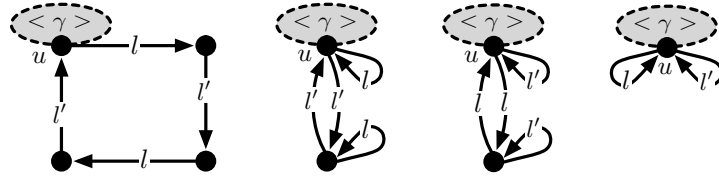
3.4.4 Contrainte de cycles

Tout d'abord étendons la contrainte de cycles aux graphes avec renommages.

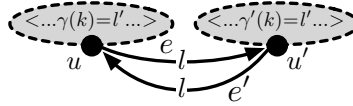
Définition 50 (contrainte de cycles sur les graphes avec renommages). Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soit G un graphe avec renommages sur la variable $x : \langle \beta \rangle$.

Soient u un sommet de G étiqueté par $\langle \gamma \rangle (x)$ et l et l' deux étiquettes d'arc de \mathcal{A}_n . La contrainte de cycles $\text{Cycle}_{u, G}(l, l')$ est satisfaite si et seulement si l'une des conditions suivantes est vérifiée :

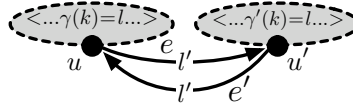
- G a un cycle $ll'll'$ incident à u ;



- G a un arc e (de source u , de cible u' et d'étiquette l), un arc inverse e' (de source u' , de cible u et d'étiquette l') et il existe un indice $k \in [1, |\beta|]$ tel que $\gamma(k) = l'$ et $\gamma'(k) = l$, où $\langle \gamma' \rangle(x)$ est l'étiquette de u' dans G ;



- réciproquement, G a un arc e (de source u , de cible u' et d'étiquette l'), un arc inverse e' (de source u' , de cible u et d'étiquette l) et il existe un indice $k \in [1, |\beta|]$ tel que $\gamma(k) = l$ et $\gamma'(k) = l'$, où $\langle \gamma' \rangle$ est l'étiquette de u' dans G ;



- l et l' sont deux étiquettes de $\Sigma_\gamma \setminus \{-\}$.



G satisfait la contrainte de cycles $Cycle_G(\mathcal{A}_n)$ si et seulement si la contrainte de cycles $Cycle_{u,G}(\alpha_i, \alpha_j)$ est satisfait pour tout sommet u de G et tout couple d'étiquettes $(\alpha_i, \alpha_j) \in \mathcal{A}_n^2$ tel que les indices i et j de $[0, n]$ vérifient $i + 2 \leq j$.

Montrons à présent sous quelle condition l'instanciation d'un graphe avec renommages satisfait la contrainte de cycles.

Proposition 51. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soit G un graphe avec renommages sur la variable $x : \langle \beta \rangle$ qui soit non orienté et vérifie : pour tout sommet v de G , il existe un ensemble \mathcal{A}_n' tel que la contrainte d'arc incident $Incid_{v,G}(\mathcal{A}_n')$ soit satisfaite.

Soit O une orbite de type $\langle \beta \rangle$ compatible avec les renommages de G , c'est-à-dire telle que pour tout couple d'indices $(k, k') \in [1, |\beta|]^2$ et toute étiquette de renommage $\langle \gamma \rangle(x)$ des sommets de G , si les étiquettes $\gamma(k) = \alpha_i$ et $\gamma(k') = \alpha_j$ vérifient $i + 2 \leq j$, alors les étiquettes $\beta(k) = \alpha_{i'}$ et $\beta(k') = \alpha_{j'}$ vérifient $i' + 2 \leq j'$.

Pour tout sommet u de G^O , et tout couple d'indices $(i, j) \in [1, n]$ tel que $i + 2 \leq j$, G satisfait la contrainte de cycles $Cycle_{g^O(u), G}(\alpha_i, \alpha_j)$, si et seulement si son instanciation G^O vérifie la contrainte de cycles $Cycle_{u, G^O}(\alpha_i, \alpha_j)$.

Démonstration. Soit u un sommet de G^O et deux étiquettes α_i et α_j de \mathcal{A}_n telles que $i + 2 \leq j$. Supposons que G^O ait un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ incident à u . Notons les arcs de ce cycle comme suit :

- c_1 l'arc de source u , de cible u_1 et d'étiquette α_i ,

- c_2 l'arc de source u_1 , de cible u_2 et d'étiquette α_j ,
- c_3 l'arc de source u_2 , de cible u_3 et d'étiquette α_i ,
- c_4 l'arc de source u_3 , de cible u et d'étiquette α_j .

Notons (v, w) le sommet de l'instanciation canonique $I_{G,O}$ isomorphe à u .

Montrons la proposition par cas sur la nature des arcs c_1 , c_2 , c_3 et c_4 qui peuvent être, chacun, issus de G ou de O

1. Supposons tout d'abord, que les quatre arcs c_1 , c_2 , c_3 et c_4 sont issus de G . Montrons alors que G a un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ incident à v .

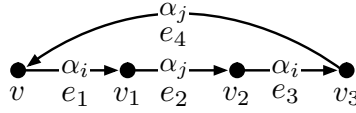
Notons e_1 l'arc de G image de c_1 par g'' , de source v , de cible v_1 et d'étiquette α_i . D'après la proposition 40, c_1 est isomorphe à l'arc (e_1, w) de l'instanciation canonique $I_{G,O}$. Et u_1 est isomorphe au sommet (v_1, w) .

Notons e_2 l'arc de G image de c_2 par g'' , de source v_1 , de cible v_2 et d'étiquette α_j . Donc c_2 est isomorphe à l'arc (e_2, w) de l'instanciation canonique $I_{G,O}$ et u_2 est isomorphe au sommet (v_2, w) .

Notons e_3 l'arc de G image de c_3 par g'' , de source v_2 , de cible v_3 et d'étiquette α_i . Donc c_3 est isomorphe à l'arc (e_3, w) de l'instanciation canonique $I_{G,O}$ et u_3 est isomorphe au sommet (v_3, w) .

Notons e_4 l'arc de G image de c_4 par g'' , de source v_3 , de cible v et d'étiquette α_j . Donc c_4 est isomorphe à l'arc (e_4, w) de l'instanciation canonique $I_{G,O}$ et u est isomorphe au sommet (v_4, w) . Or par hypothèse, u est isomorphe au sommet (v, w) . Donc $v = v_4$.

Donc G a un cycle $\alpha_i \alpha_j \alpha_i \alpha_j$ incident à v , constitué des arcs e_1 , e_2 , e_3 et e_4 .

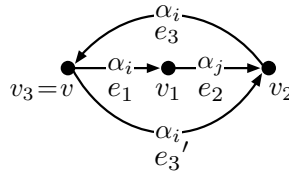


Réutilisons les mêmes notations dans les cas suivants pour les arcs issus de G .

2. Supposons que les trois arcs c_1 , c_2 et c_3 sont issus de G et que c_4 est issu de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme c_4 est issue de O , $v = v_3$. Or G est non orienté, donc e_3 a un inverse e'_3 . Et donc v est incident aux arcs e_1 et e'_3 tous deux étiquetés par α_i .

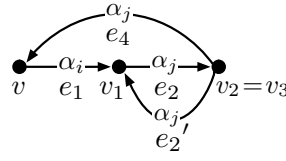
Ce qui est contradictoire avec la condition d'arcs incidents en v satisfaite par G .



3. Supposons que les trois arcs c_1 , c_2 et c_4 sont issus de G et que c_3 est issu de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme dans le cas 2, c_3 est issue de O , donc $v_2 = v_3$. Et donc, G a deux arcs incidents à v_2 étiquetés par α_j .

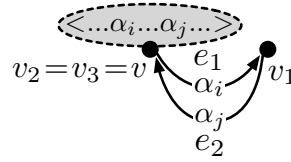
Ce qui est contradictoire avec la condition d'arcs incidents en v_2 satisfaite par G .



4. Supposons que les deux arcs c_1 et c_2 sont issus de G et que c_3 et c_4 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme c_3 et c_4 sont tous les deux issus de O , $v_2 = v_3 = v$ et l'étiquette de renommage de v dans G contient les étiquettes α_i et α_j correspondant respectivement aux étiquettes de c_3 et c_4 . Donc dans G , α_i étiquette l'arc e_1 incident au sommet v , et α_i est une lettre du renommage qui étiquette v .

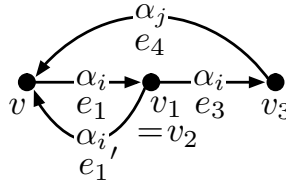
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



5. Supposons que les trois arcs c_1 , c_3 et c_4 sont issus de G et que c_2 est issu de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme dans le cas 2, c_2 est issu de O , donc $v_1 = v_2$. Et donc G a deux arcs incidents à v_1 étiquetés par α_i .

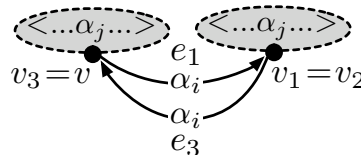
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



6. Supposons que les deux arcs c_1 et c_3 sont issus de G et que c_2 et c_4 sont issus de O . Montrons alors que G a deux arcs inverses étiquetés par α_i dont les deux sommets sources et cibles sont étiquetés par un renommage qui contient α_j .

c_2 et c_4 sont issus de O . Donc $v_2 = v_1$ et $v_3 = v$. Donc e_1 et e_3 ont respectivement pour sources v et v_1 , pour cibles v_1 et v et pour étiquette α_i . Ce sont donc 2 arcs inverses. De plus, v et v_1 sont tous les deux étiquetés par des renommages qui contiennent l'étiquette α_j , correspondant aux étiquettes de c_2 et c_4 .

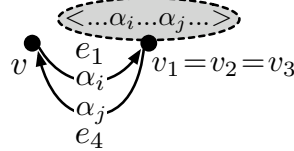
Donc G satisfait la contrainte de cycles $Cycle_{v,G}(\alpha_i, \alpha_j)$.



7. Supposons que les deux arcs c_1 et c_4 sont issus de G et que c_2 et c_3 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

c_2 et c_3 sont issus de O . Donc $v_1 = v_2 = v_3$. Comme dans la cas 4, G a son sommet v_1 qui est étiqueté par un renommage qui contient l'étiquette α_j (correspondant à l'étiquette de c_2) et un arc e_4 de source v_1 et étiqueté par α_j .

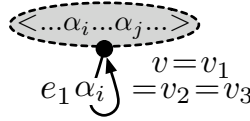
Donc G ne satisfait pas de contrainte d'arcs incidents en v_1 , ce qui est contradictoire avec les hypothèses.



8. Supposons que l'arc c_1 est issu de G et que c_2 , c_3 et c_4 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

c_2 , c_3 et c_4 sont issus de O , donc l'arc e_1 de G a le sommet v à la fois comme source et comme cible. Donc comme dans le cas 4, le sommet v de G est étiqueté par un renommage qui contient l'étiquette α_i (correspondant à l'étiquette de c_3). Et e_1 a pour source v et est étiqueté par α_i .

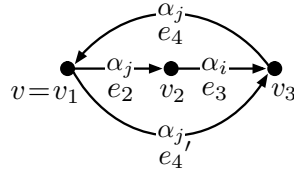
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



9. Supposons que les trois arcs c_2 , c_3 et c_4 sont issus de G et que c_1 est issu de O . Montrons que c'est contradictoire avec les hypothèses sur G .

c_1 est issu de O , donc $v = v_1$. Donc comme au cas 2, le sommet v de G est source de deux arcs étiquetés par α_j : l'arc e_2 et l'inverse de l'arc e_4 .

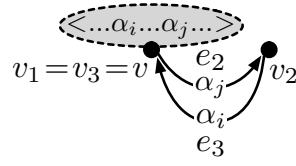
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



10. Supposons que les deux arcs c_2 et c_3 sont issus de G et que c_1 et c_4 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

c_1 et c_4 sont issus de O , donc $v = v_1$ et $v_3 = v$. Donc comme au cas 2, le sommet v de G a pour étiquette un renommage qui contient la lettre α_j (qui correspond à c_4), et est source de e_2 aussi étiqueté par α_j .

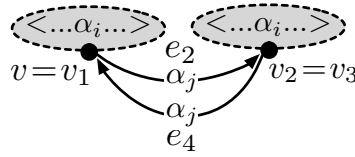
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



11. Supposons que les deux arcs c_2 et c_4 sont issus de G et que c_1 et c_3 sont issus de O . Montrons alors que G a deux arcs inverses étiquetés par α_j dont les deux sommets sources et cibles sont étiquetés par un renommage qui contient α_i .

Ce cas est le symétrique du cas 6 et pour les mêmes raisons, la propriété est vérifiée.

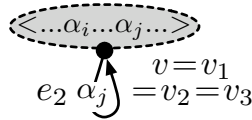
Donc G satisfait la contrainte de cycles $Cycle_{v,G}(\alpha_i, \alpha_j)$.



12. Supposons que l'arc c_2 est issu de G et les arcs c_1 , c_3 et c_4 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme pour le cas 8, l'arc e_2 de G est une boucle de source et cible v étiqueté par α_j et v est étiqueté par un renommage qui contient l'étiquette α_j (qui correspond à c_4).

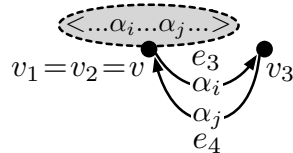
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



13. Supposons que les deux arcs c_3 et c_4 sont issus de G et que c_1 et c_2 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme pour le cas 4, le sommet v de G est étiqueté par un renommage qui contient l'étiquette α_i (qui correspond à c_1) et est la source de l'arc e_3 aussi étiqueté par α_i .

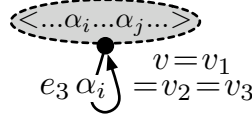
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



14. Supposons que l'arc c_3 est issu de G et les arcs c_1 , c_2 et c_4 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme pour le cas 8, l'arc e_3 de G est une boucle de source et cible v étiqueté par α_i et v est étiqueté par un renommage qui contient l'étiquette α_i (qui correspond à c_1).

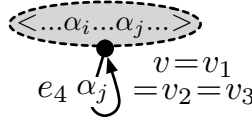
Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



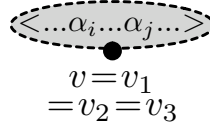
15. Supposons que l'arc c_4 est issu de G et les arcs c_1 , c_2 et c_3 sont issus de O . Montrons que cela est contradictoire avec les hypothèses sur G .

Comme pour le cas 8, l'arc e_4 de G est une boucle de source et cible v étiqueté par α_j et v est étiqueté par un renommage qui contient l'étiquette α_j (qui correspond à c_2).

Donc G ne satisfait pas de contrainte d'arcs incidents en v , ce qui est contradictoire avec les hypothèses.



16. Supposons que les 4 arcs c_1 , c_2 , c_3 et c_4 sont issus de O . Montrons que v a un renommage qui contient les deux étiquettes α_i et α_j et que ce cas n'est pas contradictoire avec les hypothèses.



Notons $\langle \gamma \rangle (x)$ le renommage qui étiquette le sommet v de G . γ contient au moins une occurrence de α_i (qui correspond à c_1 et/ou c_3) et une occurrence de α_j (qui correspond à c_2 et/ou c_4). Or G satisfait la contrainte d'arcs incidents, donc ces occurrences sont uniques. Notons $k, k' \in [1, |\beta|]$ ces indices ($\gamma(k) = \alpha_i$ et $\gamma(k') = \alpha_j$).

Or $\langle \beta \rangle$ est compatible avec $\langle \gamma \rangle$. Donc $\beta(k) = \alpha_{i'}$ et $\beta(k') = \alpha_{j'}$ vérifient $i' + 2 \leq j'$. Or O est une orbite et donc satisfait les contraintes d'arcs incidents et de cycle sur Σ_β . Donc O a un unique cycle $\alpha_{i'}\alpha_{j'}\alpha_{i'}\alpha_{j'}$ incident à w , constitué de :

- l'arc d_1 de source w , de cible w_1 et d'étiquette $\alpha_{i'}$,
- l'arc d_2 de source w_1 , de cible w_2 et d'étiquette $\alpha_{j'}$,
- l'arc d_3 de source w_2 , de cible w_3 et d'étiquette $\alpha_{i'}$,
- l'arc d_4 de source w_3 , de cible w et d'étiquette $\alpha_{j'}$.

Donc d'après la proposition 40, l'instanciation canonique $I_{G,O}$ a un cycle $\alpha_i, \alpha_j \alpha_i \alpha_j$ incident à (v, w) , constitué des arcs issus de O suivants :

- l'arc (v, d_1) , de source (v, w) , de cible (v, w_1) et d'étiquette α_i ,
- l'arc (v, d_2) , de source (v, w_1) , de cible (v, w_2) et d'étiquette α_j ,
- l'arc (v, d_3) , de source (v, w_2) , de cible (v, w_3) et d'étiquette α_i ,
- l'arc (v, d_4) , de source (v, w_3) , de cible (v, w) et d'étiquette α_j .

Or d'après la contrainte d'arcs incidents, ce cycle est unique (voir la proposition 44). Donc c_1 , c_2 , c_3 et c_4 sont respectivement isomorphes aux arcs (v, d_1) , (v, d_2) , (v, d_3) et (v, d_4) .

Donc les contraintes de cycle $Cycle_{u,GO}(\alpha_i, \alpha_j)$ et $Cycle_{v,G}(\alpha_i, \alpha_j)$ sont simultanément satisfaites dans les cas 1, 6, 11 et 16. Les autres cas sont contradictoires avec les hypothèses sur G (contrainte d'arc incidents et non orientation).

Donc $Cycle_{u,GO}(\alpha_i, \alpha_j)$ si et seulement si $Cycle_{v,G}(\alpha_i, \alpha_j)$. □

Étendons à présent la contrainte de cycles aux règles de transformation avec renommages d'orbites.

Théorème 52. Soient $n \geq 0$ une dimension et β un mot de \mathcal{A}_n^* . Soient G une pré- G -carte de dimension n , $p : L \hookleftarrow K \hookrightarrow R$ une production avec renommages sur la variable $x : < \beta >$ qui satisfait la contrainte d'arcs incidents $Incide_p(\mathcal{A}_n)$ et soit non orienté, et O une orbite de type $< \beta >$ compatible avec les renommages de p (c'est-à-dire compatible avec les renommages de L , K et R).

Si G satisfait la condition de cycle $Cycle_G(\mathcal{A}_n)$ et p satisfait la condition de cycle $Cycle_p(\mathcal{A}_n)$ définie ci-dessous, alors tout graphe topologique H résultat de la transformation directe $G \Rightarrow_{p^O, m} H$ pour un morphisme de filtrage m , satisfait la condition de cycle $Cycle_H(\mathcal{A}_n)$.

$Cycle_p(\mathcal{A}_n)$: la contrainte de cycles $Cycle_p(\mathcal{A}_n)$ est satisfaite, si et seulement si pour tout couple d'indices $(i, j) \in [1, n]$ qui vérifie $i + 2 \leq j$, les trois conditions suivantes sont satisfaites :

1. soit v un sommet conservé de K , si L satisfait la contrainte de cycles $Cycle_{v,L}(\alpha_i, \alpha_j)$, alors R satisfait la condition de cycle $Cycle_{v,R}(\alpha_i, \alpha_j)$;
2. soit v un sommet ajouté de $R \setminus K$, R satisfait la condition de cycle $Cycle_{v,R}(\alpha_i, \alpha_j)$;
3. soit v un sommet conservé de K , si L ne satisfait pas la contrainte de cycles $Cycle_{v,L}(\alpha_i, \alpha_j)$, alors tout arc e de L incident à v étiqueté par α_i ou α_j , admet un arc isomorphe dans R .

Démonstration. Supposons que p satisfait la contrainte de cycles $Cycle_p(\mathcal{A}_n)$. Montrons d'abord, que l'instance p^O satisfait la contrainte de cycles $Cycle_{p^O}(\mathcal{A}_n)$. Soit un couple d'indices $(i, j) \in [1, n]$ qui vérifie $i + 2 \leq j$

1. Soit u un sommet conservé de K^O isomorphe au sommet (v, w) de l'instanciation canonique $I_{K,O}$, tel que L^O satisfait la contrainte de cycles $Cycle_{u,L^O}(\alpha_i, \alpha_j)$.
D'après la proposition 51 précédente, L satisfait la contrainte de cycles $Cycle_{v,L}(\alpha_i, \alpha_j)$. Or par construction, v est un sommet de K . Donc le premier point de la contrainte de cycles sur p s'applique, et donc R satisfait la contrainte de cycles $Cycle_{v,R}(\alpha_i, \alpha_j)$. Or d'après la proposition 51, R^O satisfait la contrainte de cycles $Cycle_{u,R^O}(\alpha_i, \alpha_j)$.
Donc le premier point de la contrainte de cycles est satisfait pour p^O .
2. Soit u un sommet ajouté de $R^O \setminus K^O$ isomorphe au sommet (v, w) de l'instanciation canonique $I_{K,O}$.
D'après la proposition 40, v est un sommet de $R \setminus K$. Donc le deuxième point de la contrainte de cycles sur p s'applique, et donc R satisfait la contrainte de cycles $Cycle_{v,R}(\alpha_i, \alpha_j)$. Or d'après la proposition 51, R^O satisfait la contrainte de cycles $Cycle_{u,R^O}(\alpha_i, \alpha_j)$.
Donc le deuxième point de la contrainte de cycles est satisfait pour p^O .
3. Soit u un sommet conservé de K^O isomorphe au sommet (v, w) de l'instanciation canonique $I_{K,O}$, tel que L^O ne satisfait pas la contrainte de cycles $Cycle_{u,L^O}(\alpha_i, \alpha_j)$.
D'après la proposition 51, L ne satisfait pas non plus la contrainte de cycles $Cycle_{v,L}(\alpha_i, \alpha_j)$. Donc d'après le troisième point de la contrainte de cycles sur p , tout arc e de L incident à v étiqueté par α_i ou α_j , admet un arc isomorphe f dans R . Donc d'après la proposition 40, L^O a un arc isomorphe à l'arc (e, w) de l'instanciation canonique $I_{L,O}$, qui a pour source u et la même étiquette α_i ou α_j que e . De même, R^O a un arc isomorphe à l'arc (f, w) de l'instanciation canonique $I_{R,O}$, qui a pour source u et la même étiquette α_i ou α_j que f (et donc e). Or e et f sont isomorphes, donc (e, w) et (f, w) aussi.
Donc le troisième point de la contrainte de cycles est satisfait pour p^O .

Donc la production instanciée p^O satisfait la contrainte de cycles $Cycle_{p^O}(\mathcal{A}_n)$.

Donc d'après la proposition 19, page 46 tout graphe topologique H résultat de la transformation directe $G \Rightarrow_{p^O, m} H$ pour un morphisme de filtrage m , satisfait la condition de cycle $Cycle_H(\mathcal{A}_n)$. \square

Illustrons la proposition précédente à l'aide de l'opération de cône. Rappelons que la règle de cône p_{cone} ainsi que son instanciation p_{cone}^O par l'orbite O de type $< \alpha_0 \alpha_1 >$ sont illustrées en Fig. 3.30 page 88. Remarquons tout d'abord que l'orbite O est compatible avec les renommages qui étiquettent les sommets de p_{cone} . En effet, il n'existe pas de couple d'étiquettes $(\alpha_i, \alpha_j) \in \Sigma_{\alpha_0 \alpha_1}^2$ tel que $i + 2 \leq j$.

La règle p_{cone} satisfait la contrainte de cycles $Cycle_{p_{cone}}(\mathcal{A}_3)$. Illustrons le premier point de la contrainte de cycles des règles avec renommages d'orbites. Considérons l'unique sommet 1 de K_{cone} , ce sommet satisfait les contraintes de cycles $Cycle_{1, L_{cone}}(\alpha_0, \alpha_3)$ (voir définition 50 page 94). Ce sommet entre dans le troisième cas de la définition 50, il possède une boucle incidente étiquetée par α_3 et il existe une occurrence de l'étiquette α_0 dans le renommage $< \alpha_0 \alpha_1 >$ qui étiquette le sommet 1. D'après le premier point de la contrainte de cycles exprimée dans la proposition précédente, le sommet 1 doit également satisfaire $Cycle_{1, R_{cone}}(\alpha_0, \alpha_3)$. Ainsi, dans la partie droite R_{cone} , le sommet 1 entre également dans le troisième cas dans la définition 50. En effet, dans R_{cone} , il existe deux arcs étiquetés par α_3 d'orientation inverse qui connectent les sommets 1 et 2. De plus, l'occurrence de l'étiquette α_0 est à la même position dans les renommages qui étiquettent les sommets 1 et 2. Remarquons également que les contraintes $Cycle_{1, L_{cone}}(\alpha_1, \alpha_3)$ et $Cycle_{1, R_{cone}}(\alpha_1, \alpha_3)$ sont également respectées dans p_{cone} .

Illustrons à présent le second point de la contrainte de cycles des règles avec renommages d'orbites. Considérons le sommet ajouté 4 de $R_{cone} \setminus K_{cone}$. D'après la proposition précédente, ce sommet doit satisfaire les contraintes $Incide_{4, R_{cone}}(\alpha_0, \alpha_2)$, $Cycle_{4, L_{cone}}(\alpha_1, \alpha_3)$ et $Cycle_{4, L_{cone}}(\alpha_0, \alpha_3)$. La satisfaction de $Cycle_{4, L_{cone}}(\alpha_0, \alpha_2)$ et $Cycle_{4, L_{cone}}(\alpha_1, \alpha_3)$ est à nouveau donnée par le troisième cas de la définition 50. En revanche, la satisfaction de $Cycle_{4, L_{cone}}(\alpha_0, \alpha_3)$ est donnée par le premier cas de la définition. En effet, il existe dans R_{cone} un cycle $\alpha_0 \alpha_3 \alpha_0 \alpha_3$ incident au sommet 4.

Enfin, l'exemple du cône n'est pas pertinent pour illustrer le dernier point de la contrainte de cycles des règles avec renommages d'orbites. En effet, seule la contrainte $Cycle_{1, L_{cone}}(\alpha_0, \alpha_2)$ n'est pas satisfaite dans L_{cone} et le sommet 1 ne possède ni arc incident étiqueté par α_0 , ni arc incident étiqueté par α_2 .

Ainsi, la contrainte de cycle $Incide_{p_{cone}}(\mathcal{A}_3)$ est satisfaite et conformément à la proposition précédente, l'instanciation p_{cone}^O de p_{cone} par l'orbite O satisfait également la contrainte $Cycle_{p_{cone}^O}(\mathcal{A}_n)$ introduite dans la proposition 19 page 46. Remarquons notamment que le sommet $(1, a)$ de K_{cone}^O satisfait les contraintes $Cycle_{(1, a), L_{cone}^O}(\alpha_0, \alpha_3)$ et $Cycle_{(1, a), R_{cone}^O}(\alpha_0, \alpha_3)$. En effet, dans L_{cone}^O il existe un cycle $\alpha_1 \alpha_3 \alpha_1 \alpha_3$ incident à $(1, a)$. Les arcs étiquetés par α_3 de ce cycle sont issus de la boucle α_3 incidente à 1 dans L_{cone} tandis que les arcs α_0 du cycle sont issus du renommage identité des arcs étiquetés par α_0 qui connectent les sommets a et b dans O . De la même manière, pour ce même sommet $(1, a)$, les contraintes $Cycle_{(1, a), L_{cone}^O}(\alpha_1, \alpha_3)$ et $Cycle_{(1, a), R_{cone}^O}(\alpha_1, \alpha_3)$ sont simultanément satisfaites. Enfin, tous les sommets ajoutés de $R_{cone}^O \setminus K_{cone}^O$ possèdent bien tous leurs cycles. Par exemple, le sommet $(5, a)$ possède ses trois cycles incidents étiquetés par $\alpha_0 \alpha_2 \alpha_0 \alpha_2$, $\alpha_0 \alpha_3 \alpha_0 \alpha_3$ et $\alpha_1 \alpha_3 \alpha_1 \alpha_3$. Les arcs étiquetés par α_3 de ces cycles sont issus des boucles α_3 incidentes aux sommets 5 et 4 dans R_{cone} tandis que les autres arcs sont issus des renommages des arcs de O .

Illustrons à présent brièvement le troisième point de la contrainte de cycles des règles avec renommages d'orbites à l'aide de l'opération de chanfreinage (voir Fig. 3.31 page 89). Remarquons tout d'abord que comme précédemment, l'orbite O_s est compatible avec les renommages qui étiquettent les sommets de p_{chan} . Intéressons-nous à l'unique sommet 1 de L_{chan} . Ce sommet ne satisfait pas la contrainte de

cycles $Cycle_{1,L_{chan}}(\alpha_0, \alpha_3)$. Ainsi, en vertu de la contrainte de cycles introduite dans la proposition précédente, il doit exister dans R_{chan} la même boucle étiquetée par α_3 et incidente au sommet 1 dans L_{chan} . Cette boucle existe bien dans R_{chan} . Par conséquent, dans l'instanciation $p_{chan}^{O_s}$ de p_{chan} par l'orbite O_s , tous les sommets de la copie O_1 de O_s ne satisfont pas non plus la contrainte de cycles, mais ont une boucle incidente étiquetée par α_3 .

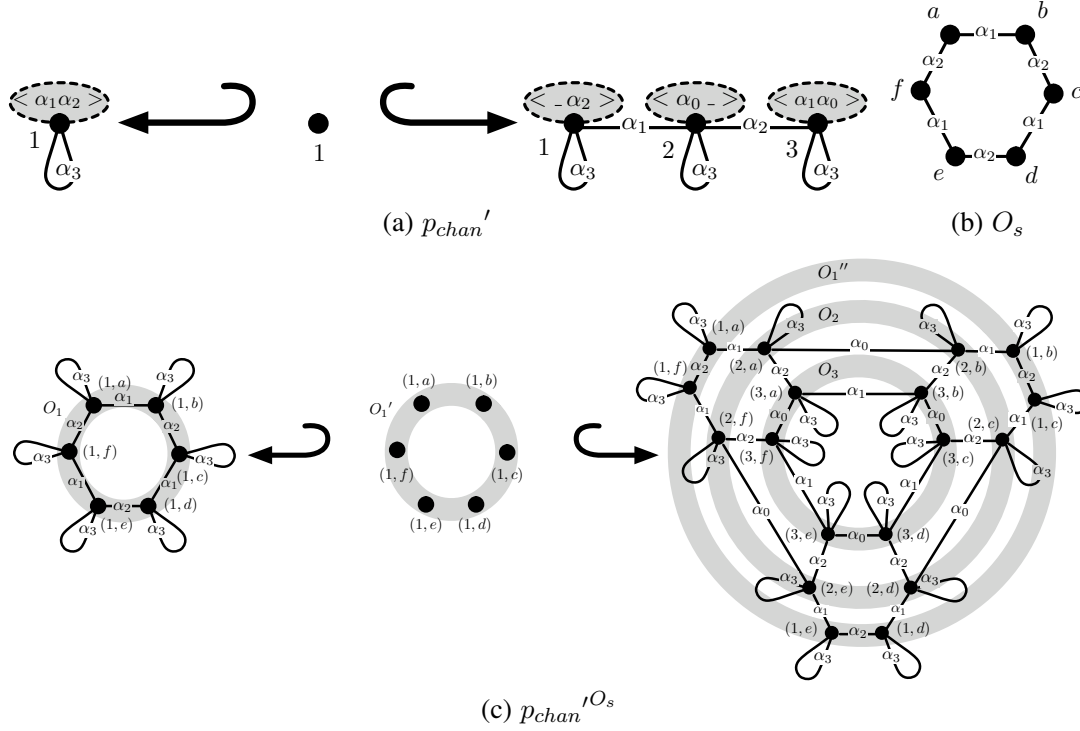


FIGURE 3.33 – Instanciation $p_{chan}^{O_s}$ de la règle p_{chan}' par une orbite O_s

Prenons enfin un exemple de règle avec renommages d'orbites qui ne respecte pas la contrainte de cycles. Considérons la règle p_{chan}' illustrée en Fig. 3.33(a). Cette règle est proche de la règle du chanfreinage de la Fig. 3.31(a) à ceci près que α_0 et α_1 ont été intervertis dans le renommage qui étiquette le sommet 3. La règle p_{chan}' ne satisfait pas la contrainte de cycles du théorème 52. En effet, les sommets 2 et 3 – qui sont des sommets ajoutés de $R_{chan}' \setminus K_{chan}'$ ne satisfont pas les contraintes $Cycle_{2,R_{chan}'}(\alpha_0, \alpha_2)$ et $Cycle_{3,R_{chan}'}(\alpha_0, \alpha_2)$. La violation de ces contraintes est due aux occurrences de l'étiquette α_0 , qui ne se trouvent pas aux mêmes positions dans les renommages qui étiquettent les sommets 2 et 3 (voir le troisième cas de la contrainte de cycles des graphes avec renommages, introduite dans la définition 50 page 94).

L'instanciation $p_{chan}^{O_s}$ de la règle p_{chan}' par l'orbite O_s de la Fig. 3.33(b) est illustrée en Fig. 3.33(c). On remarque notamment que le sommet ajouté (3, a) de $R_{chan}^{O_s} \setminus K_{chan}^{O_s}$, la violation de la contrainte de cycles $Cycle_{3,R_{chan}'}(\alpha_0, \alpha_2)$ entraîne une violation de la contrainte de cycle $Cycle_{(3,a),R_{chan}^{O_s}}(\alpha_0, \alpha_2)$. En effet, il n'existe pas de cycle $\alpha_0 \alpha_2 \alpha_0 \alpha_2$ qui passe par le sommet (3, a). Par conséquent, la contrainte de cycles $Cycle_{p_{chan}^{O_s}}(\mathcal{A}_3)$ n'est pas respectée.

Nous pouvons enfin écrire la condition qui nous assure qu'une règle de transformation avec renommages d'orbites transforme une G -carte en une G -carte.

Théorème 53. Soient $n \geq 0$ une dimension et β un mot sur \mathcal{A}_n^* . Soient G une G -carte de dimension n , $p : L \hookleftarrow K \hookrightarrow R$ une production avec renommages sur la variable $x : \langle \beta \rangle$ qui satisfait les contraintes d'arcs incidents $\text{Incede}_p(\mathcal{A}_n)$, de cycle $\text{Cycle}_p(\mathcal{A}_n)$, et soit non orientée, et O une orbite de type $\langle \beta \rangle$ compatible avec les renommages de p (c'est-à-dire compatible avec les renommages de L , K et R).

Tout graphe topologique H , résultat de la transformation directe $G \Rightarrow_{p^O, m} H$ pour un morphisme de filtrage m existe et est une G -carte de dimension n .

Démonstration. Ceci découle directement du corollaire 48, sur la conservation des pré- G -cartes, du théorème 49 sur la condition d'arcs pendants et du théorème 52 précédent, sur la préservation de la contrainte de cycles.

□

PERSPECTIVES : TRANSFORMATION DE GRAPHES ET CARTES GÉNÉRALISÉES PLONGÉES

Sommaire

4.1	Modélisation des jonctions communicantes	106
4.2	Cartes généralisées plongées	107
4.3	Variables attributs et plongement	108
4.3.1	Règles de modification de plongement	108
4.3.2	Préconditions de plongement	110
4.3.3	Contrainte de plongement	111
4.4	De l'utilité de manipuler plusieurs variables orbites	112

Le modèle des cartes généralisées présenté dans le chapitre 1 permet de modéliser la topologie des objets en s'abstrayant de leur forme, c'est-à-dire de leur géométrie. Cependant, dans une application, il est peu pertinent d'oublier totalement la forme des objets que l'on modélise. Par exemple, dans un modèleur géométrique [Ble , May , 3ds], il est nécessaire d'associer une géométrie aux objets afin de les visualiser. Pour cela, on attache des données – notamment géométriques – aux cellules topologiques. Dans la littérature, on dit que l'on *plonge une structure topologique*. De la même manière, le *plongement* désigne les données que l'on associe à la structure topologique.

Dans le chapitre précédent, nous avons introduit des règles de transformations avec variables qui permettent de transformer des G-cartes topologiques, c'est-à-dire dénuées de plongement. Dans ce chapitre de perspectives – qui clôt la première partie de ce manuscrit – nous évoquons quelques pistes afin d'étendre nos règles aux cartes généralisées plongées. En section 4.1, nous introduisons un petit mécanisme biologique, les jonctions communicantes, qui nous permet notamment d'introduire les plongements que nous prenons pour exemple dans ce chapitre. En section 4.2, nous présentons les cartes généralisées plongées et évoquons notamment une nouvelle contrainte dite *de plongement*, qui vient s'ajouter aux contraintes d'arcs incidents et de cycles, étudiées dans les chapitres précédents. En section 4.3, nous montrons comment les variables attributs définies par B. Hoffmann [Hoffmann 05] nous permettent d'écrire des règles de transformation qui modifient le plongement ainsi que des conditions de plongement qui contraignent l'application de nos règles. Enfin, en section 4.4 nous étendons notre formalisme en proposant de nouvelles règles dans lesquelles sont mises en jeu plusieurs variables orbites. Nous positionnons également ces nouvelles règles par rapport à la contrainte de plongement.

4.1 Modélisation des jonctions communicantes

Le mécanisme des jonctions communicantes [Alberts 02] est bien connu en biologie. Lorsque deux cellules biologiques sont collées l'une à l'autre le long de leur membrane, de petits canaux – les jonctions communicantes – permettent de faire circuler certaines protéines d'une cellule à l'autre. En Fig. 4.1, nous proposons une modélisation en trois étapes de ce mécanisme. Notre modélisation, bien que volontairement simplifiée, est suffisante pour illustrer ce chapitre.

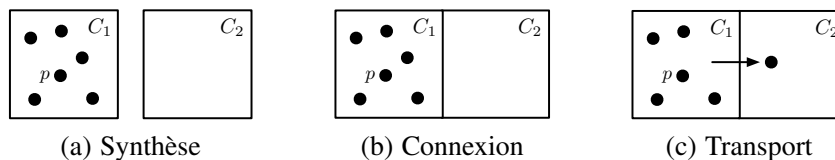


FIGURE 4.1 – Mécanisme des jonctions communicantes

Lors de l'étape de *synthèse* (voir Fig. 4.1(a)), deux cellules biologiques C_1 et C_2 sont immergées dans un même milieu. Dans la cellule C_1 , des protéines p – représentées par des points noirs – sont synthétisées. Intuitivement, à chaque instant, de nouvelles protéines p apparaissent dans C_1 . Remarquons que lors de l'étape de synthèse, C_1 et C_2 sont déconnectées. Leur connexion s'effectue dans une seconde étape, que nous nommons l'étape de *connexion* (voir Fig. 4.1(b)). Nous supposons que la connexion a lieu lorsque les cellules C_1 et C_2 sont suffisamment proches l'une de l'autre. Enfin, lors de l'étape de *transport* (voir Fig. 4.1(c)), des protéines p de C_1 sont transportées vers C_2 via des jonctions communicantes, implicites dans notre modélisation.

Supposons à présent que nous modélisons le mécanisme des jonctions communicantes à l'aide d'une carte généralisée de dimension ¹ 2. Dans ce cas, nous choisissons de représenter les cellules biologiques

1. Pour une modélisation plus réaliste, il faudrait choisir la dimension 3. Cependant, la dimension 2 simplifie les différentes figures qui illustrent ce chapitre, sans diminuer la généralité de notre discours.

C_1 et C_2 par deux faces. La connexion de C_1 et de C_2 est alors modélisée par la 2-couture des deux faces. Deux données de plongement nous permettent de spécialiser notre 2-G-carte pour cet exemple biologique. Afin de positionner nos cellules biologiques dans l'espace – notamment pour tester si elles sont suffisamment proches pour l'étape de connexion – nous associons un couple de coordonnées 2D à chaque 0-cellule (rappelons que les 0-cellules d'une carte généralisée modélisent les sommets topologiques des objets). Nous appelons ce premier plongement le *plongement géométrique*. Il s'agit en fait d'un plongement polyédrique, déjà mentionné en section 1.2. Un second plongement, que nous appelons le *plongement biochimique*, nous permet de modéliser les protéines p contenues dans C_1 et C_2 . Pour cela, nous associons une quantité de protéines aux 2-cellules de notre carte généralisée (rappelons que les 2-cellules d'une carte généralisée modélisent les faces topologiques des objets).

4.2 Cartes généralisées plongées

Rigoureusement, plonger une structure topologique signifie associer des données aux différentes cellules topologiques qui composent cette structure ou plus généralement aux différentes orbites. Nous savons que dans le modèle des cartes généralisées, il n'existe pas de représentation explicite des cellules topologiques. En effet, les cellules sont représentées par des assemblages particuliers de brins (voir les notions d'orbite et de i -cellule introduites en section 1.3.1). Ce sont donc les brins qui doivent porter les plongements. Dans la pratique, les plongements étiquettent les brins.

Dans certaines représentations des cartes généralisées plongées, un seul brin par orbite plongée porte le plongement. Dans notre cas, les orbites plongées sont les 0-cellules (c'est-à-dire les orbites sommets de type $\langle \alpha_1 \alpha_2 \rangle$) et les 2-cellules (c'est-à-dire les orbites faces de type $\langle \alpha_0 \alpha_1 \rangle$). Nous choisissons une seconde convention dans laquelle on impose que chaque brin d'une orbite plongée porte le plongement. Ceci implique que chaque brin d'une même orbite plongée porte le même plongement. Nous appelons cette propriété la *contrainte de plongement*. Notons que sous couvert de la contrainte de plongement, ces deux modes de représentation d'une carte généralisée plongée sont équivalents. Nous définissons alors les cartes généralisées plongées comme étant des cartes généralisées qui satisfont la contrainte de plongement.

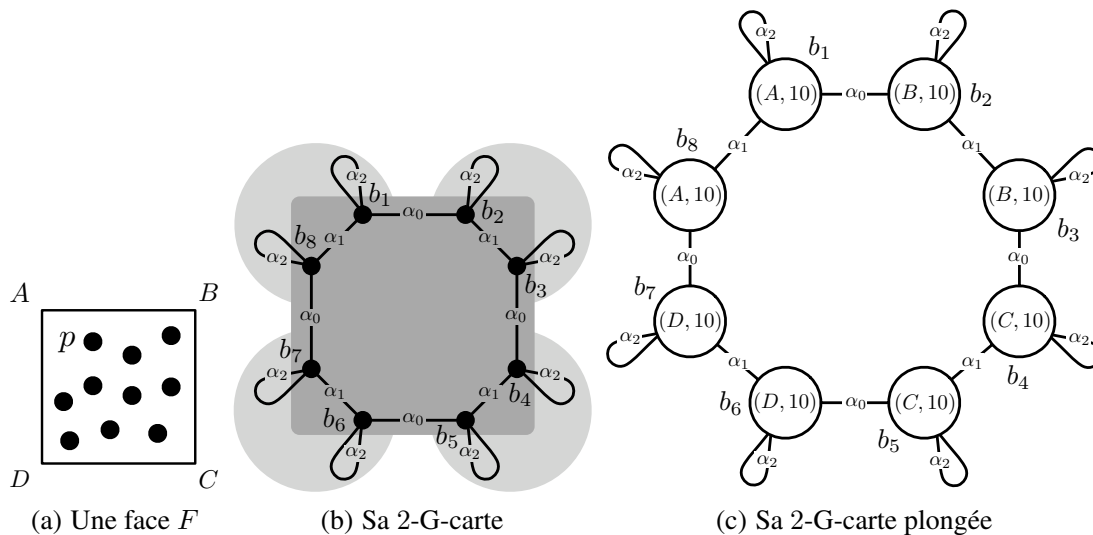


FIGURE 4.2 – Modélisation d'une face par une 2-G-carte plongée

Illustrons la notion de carte généralisée plongée à l'aide de l'exemple de la Fig. 4.2. Nous nous intéressons à une face carrée F dont les quatre sommets topologiques sont les points A , B , C et D et la

quantité de protéines p est de 10 (voir Fig. 4.2(a)). Sa topologie est représentée par la 2-G-carte – non plongée – proposée en Fig. 4.2(b). La face F est modélisée par l'orbite $\langle \alpha_0\alpha_1 \rangle (b_1)$. Les sommets topologiques A, B, C et D sont représentés respectivement par les orbites $\langle \alpha_1\alpha_2 \rangle (b_1)$, $\langle \alpha_1\alpha_2 \rangle (b_2)$, $\langle \alpha_1\alpha_2 \rangle (b_3)$ et $\langle \alpha_1\alpha_2 \rangle (b_4)$. Remarquons que chaque brin est muni d'une boucle incidente étiquetée par α_2 . Ceci signifie que cette 2-G-carte modélise une face isolée, c'est-à-dire qui n'est collée à aucune autre face. Elle permet notamment de modéliser la cellule biologique C_1 lors de l'étape de synthèse (voir Fig 4.1(a)). On remarque également que les brins ne sont pas étiquetés puisqu'ils ne portent pas encore de plongement.

En Fig. 4.2(c) les brins sont étiquetés par les plongements, il s'agit donc d'une 2-G-carte plongée. Étudions le brin b_1 , comme celui-ci appartient à l'orbite sommet $\langle \alpha_1\alpha_2 \rangle (b_1)$ et à l'orbite face $\langle \alpha_0\alpha_1 \rangle (b_1)$, il porte donc un plongement géométrique associé au sommet et un plongement biochimique associé à la face. Nous étiquetons alors b_1 par un couple $(A, 10)$. Dans ce couple, A est le plongement géométrique (il s'agit d'un point) tandis que 10 est le plongement biochimique (il s'agit d'une quantité de protéines). En vertu de la contrainte de plongement, on remarque que les brins b_2 et b_3 – qui appartiennent tous les deux à la même orbite sommet de type $\langle \alpha_1\alpha_2 \rangle$ – portent le même plongement géométrique A . De la même manière, comme les brins b_1 à b_8 appartiennent tous à la même orbite face de type $\langle \alpha_0\alpha_1 \rangle$, ils portent tous le même plongement biochimique 10.

Notons que dans la littérature, il n'existe pas de définition rigoureuse générale des cartes généralisées plongées. En effet, le plus souvent, les définitions proposées considèrent un plongement particulier lié à une application donnée. Par exemple, dans [Bellet 08], les auteurs définissent rigoureusement des cartes généralisées plongées géométriquement en s'appuyant sur les algèbres géométriques [Charneau 07]. Nous proposons une autre approche où l'on considère que les plongements qui étiquettent les brins sont des n -uplets de valeurs typées, ou plus généralement des expressions typées. Dans notre exemple, l'ensemble Σ_V des étiquettes des brins est un ensemble de paires, dont le premier élément représente un point géométrique et le second une quantité de protéines. De plus, comme nous l'avons vu, chaque plongement est associé à une orbite de type $\langle \beta \rangle$. Notons $\Pi \langle \beta \rangle$ la projection de Σ_V associée à $\langle \beta \rangle$. Dans notre cas, comme nous plongeons les orbites de types $\langle \alpha_1\alpha_2 \rangle$ et $\langle \alpha_0\alpha_1 \rangle$, les plongements sont définis respectivement par $\Pi \langle \alpha_1\alpha_2 \rangle$ et $\Pi \langle \alpha_0\alpha_1 \rangle$. Par exemple, pour le brin b_1 , nous avons $(\Pi \langle \alpha_1\alpha_2 \rangle)(b_1) = A$ et $(\Pi \langle \alpha_0\alpha_1 \rangle)(b_1) = 10$.

4.3 Variables attributs et plongement

Dans cette section, nous montrons comment la combinaison des variables attributs définies par B. Hoffmann [Hoffmann 05] et de nos variables orbites nous permettent de définir des règles de transformation qui modifient les plongements d'une carte généralisée (voir section 4.3.1). Nous proposons également de définir des conditions de plongement qui contraignent l'application de nos règles (voir section 4.3.2). Celles-ci nous permettent notamment de traduire par une règle l'étape de connexion (voir Fig. 4.1(b)), qui rappelons-le, ne peut se produire que si les deux cellules biologiques C_1 et C_2 sont suffisamment proches l'une de l'autre. Enfin, en section 4.3.3, nous montrons comment la contrainte de plongement introduite dans la section précédente s'étend aux règles de modification de plongement.

4.3.1 Règles de modification de plongement

Les variables attributs de B. Hoffmann sont les premières que nous avons présentées en section 3.1 page 50, afin de donner l'intuition des règles de transformation avec variables. Nous pouvons les utiliser telles quelles afin d'écrire des règles de transformation qui modifient les plongements. Intuitivement, dans une telle règle de transformation, les sommets sont étiquetés par des variables attributs, ou plus généralement des expressions construites sur ces variables. Dans la suite, nous parlons de *variables de*

plongement plutôt que de variables attributs.

Intéressons-nous à l'étape de synthèse du mécanisme des jonctions communicantes (voir Fig. 4.1(a)). Lors de la synthèse, nous devons filtrer une cellule biologique isolée, c'est à dire une orbite de type $\langle \alpha_0 \alpha_1 \rangle$ isolée, puis augmenter sa quantité de protéines. En termes de règle de transformation de 2-G-cartes, il s'agit dans un premier temps de filtrer une orbite de type $\langle \alpha_0 \alpha_1 \rangle$ dont tous les brins possèdent une boucle incidente étiquetée par α_2 , puis dans un second temps de mettre à jour le plongement biochimique associé.

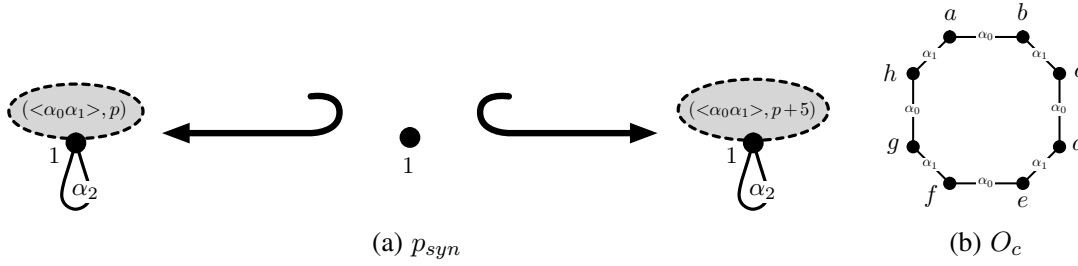


FIGURE 4.3 – La règle de synthèse p_{syn} et l'orbite carrée O_c

Nous proposons alors de traduire l'étape de synthèse par la production $p_{syn} : L_{syn} \leftarrow K_{syn} \hookrightarrow R_{syn}$ illustrée en Fig. 4.3(a). Nous l'appelons la *règle de synthèse*. La production p_{syn} contient à la fois le renommage d'orbite $\langle \alpha_0 \alpha_1 \rangle$ (x), la variable orbite $x : \langle \alpha_0 \alpha_1 \rangle$ et la variable de plongement biochimique p . Rappelons que conformément aux morphismes de graphes partiellement étiquetés (voir annexe A.2, définition 57), afin de modifier le plongement d'un brin (c'est-à-dire afin de modifier l'étiquette d'un brin), ce dernier ne doit pas être étiqueté dans l'interface de la production. C'est pourquoi le brin 1 n'est pas étiqueté dans K_{syn} .

Dans L_{syn} , le sommet 1 est étiqueté par le couple $(\langle \alpha_0 \alpha_1 \rangle, p)$. Ceci signifie que nous filtrons une orbite de type $\langle \alpha_0 \alpha_1 \rangle$ de plongement biochimique p , c'est-à-dire une cellule biologique qui contient p protéines. Dans la partie droite de la règle, le sommet 1 est étiqueté par le couple $(\langle \alpha_0 \alpha_1 \rangle, p+5)$, ce qui signifie que l'orbite filtrée par L_{syn} est renommée par l'identité $\langle \alpha_0 \alpha_1 \rangle : \langle \alpha_0 \alpha_1 \rangle \rightarrow \langle \alpha_0 \alpha_1 \rangle$, sa topologie n'est donc pas modifiée. Dans l'expression $p+5$, p est la variable de plongement biochimique filtrée par L_{syn} et 5 est la quantité de protéines ajoutées lors d'une étape de synthèse. Lors de l'étape de synthèse, p est substituée par sa valeur dans l'objet filtré, puis $p+5$ est évaluée pour calculer le nouveau plongement.

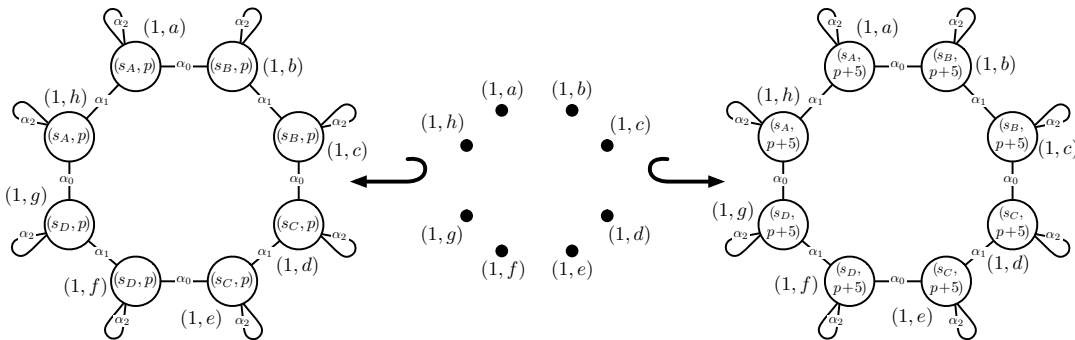


FIGURE 4.4 – $p_{syn}^{O_c}$

L'instanciation $p_{syn}^{O_c}$ de la règle de synthèse par l'orbite carrée O_c (voir Fig. 4.3(b)) est illustrée en Fig. 4.4. Le principe est analogue à l'instanciation des productions avec renommages (voir section 3.3.5 page 77) à ceci près que les sommets de $L_{syn}^{O_c}$ et $R_{syn}^{O_c}$ sont à présent étiquetés par des variables de plongement. Rappelons que les 2-G-cartes que nous manipulons dans le cadre de notre exemple présentent deux orbites plongées : les orbites sommets de type $\langle \alpha_1 \alpha_2 \rangle$ et les orbites faces de type $\langle \alpha_0 \alpha_1 \rangle$. Ainsi, comme dans la 2-G-carte plongée de la Fig. 4.2(c), les sommets de $L_{syn}^{O_c}$ sont étiquetés par un couple dont le premier membre est une variable de plongement géométrique et le second membre est une variable de plongement biochimique.

Les variables de plongement biochimiques des sommets de $p_{syn}^{O_c}$ sont attribués de la manière suivante. La variable de plongement biochimique p associée au sommet 1 dans la partie gauche de la règle de synthèse est reportée sur tous les sommets abstraits par le sommet 1, c'est-à-dire dans tous les sommets de $L_{syn}^{O_c}$. De la même manière, l'expression de plongement biochimique $p + 5$ de la partie droite de la règle de synthèse est reportée sur tous les sommets de $R_{syn}^{O_c}$.

Comme la règle de synthèse ne manipule que des plongements biochimiques, les expressions de plongement géométrique de $R_{syn}^{O_c}$ sont réduites aux variables de plongement géométrique. Ainsi, dans $L_{syn}^{O_c}$, on remarque que les brins $(1, a)$ et $(1, h)$ – qui appartiennent à la même orbite sommet de type $\langle \alpha_0 \alpha_1 \rangle$ – portent une même variable de plongement géométrique s_A . Ils sont également étiquetés par s_A dans $R_{syn}^{O_c}$. Ainsi, la production $p_{syn}^{O_c}$ ne modifie pas les plongements géométriques des sommets topologiques filtrés. La production $p_{syn}^{O_c}$ nous permet donc d'augmenter la quantité de protéines d'une cellule biologique modélisée par une face carrée sans modifier la position de ses sommets.

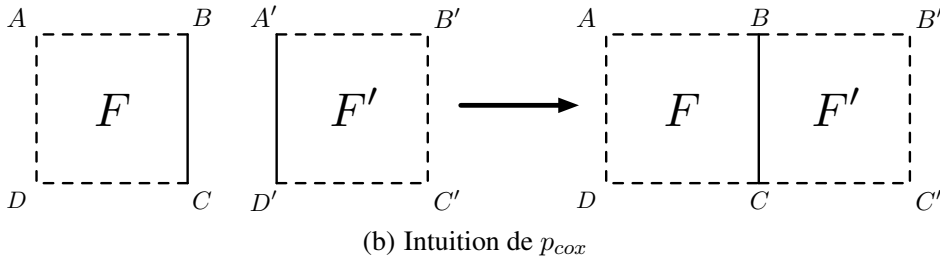
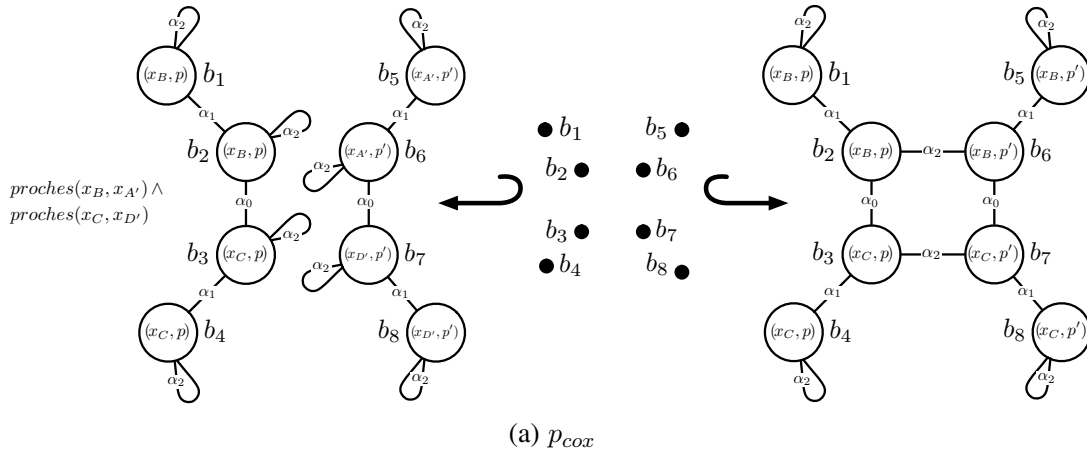
Remarquons que par souci de pédagogie, nous venons de présenter l'instanciation de la variable orbite dans la règle de synthèse. Nous avons ainsi construit une nouvelle règle $p_{syn}^{O_c}$ qui ne comprend plus de variable orbite mais qui comprend encore les variables de plongement de p_{syn} . À leur tour, les variables de plongement peuvent être instanciées de la même manière que les variables attributs de B. Hoffmann [Hoffmann 05]. En pratique, nous devons effectuer ces deux étapes d'instanciation – des variables orbites et des variables de plongement – simultanément.

4.3.2 Préconditions de plongement

Afin d'introduire les préconditions de plongement, plaçons-nous dans le cadre de l'étape de connexion du mécanisme des jonctions communicantes (voir Fig. 4.1(b)). Au cours de cette étape, lorsque les cellules biologiques C_1 et C_2 sont suffisamment proches, elles se connectent l'une à l'autre le long de leur membrane. Les variables attributs de Hoffmann nous permettent d'écrire une règle qui ne s'applique que lorsque cette précondition est satisfaite.

Ainsi, nous traduisons l'étape de connexion en la règle avec variables de plongement p_{cox} illustrée en Fig. 4.5(a). Nous l'appelons la *règle de connexion*. Six variables de plongement apparaissent dans p_{cox} : $p, p', x_B, x_C, x_{A'}$ et $x_{D'}$. Remarquons que cette règle ne comporte pas de variable orbite. L'intuition de la règle est donnée en Fig. 4.5(b). Du point de vue purement topologique, comme nous représentons les cellules biologiques par deux orbites faces de type $\langle \alpha_0 \alpha_1 \rangle$, leur connexion se traduit par une 2-couture le long de leurs arêtes topologiques. Intuitivement, nous filtrons deux arêtes topologiques – représentées en traits pleins sur la Fig. 4.5(b) – et les rassemblons en une seule arête topologique dans la partie droite de la règle. Remarquons que les plongements biochimiques ne sont pas modifiés car les quantités de protéines n'évoluent pas au cours de l'étape de connexion. En revanche, nous devons mettre à jour le plongement géométrique. Ainsi, suite à l'étape de connexion, les sommets topologiques B et A' – de même que C et D' – doivent être à la même position.

Ainsi, dans la partie gauche L_{cox} de la règle de connexion, nous filtrons deux arêtes topologiques du bord et leurs sommets respectifs. La première est composée des brins b_1 à b_4 munis de leurs arcs inci-

FIGURE 4.5 – La règle de connexion p_{cox}

dents, elle modélise l'arête topologique BC illustrée dans la partie gauche de la Fig. 4.5(b). La seconde est composée des brins b_5 à b_8 munis de leurs arcs incidents, elle modélise l'arête topologique $A'D'$ de la Fig. 4.5(b). Dans la partie droite R_{cox} , les deux faces topologiques sont 2-cousues le long de deux arêtes topologiques. Comme les brins b_1, b_2, b_5 et b_6 appartiennent désormais au même sommet topologique, la variable de plongement géométrique x_B est recopiée dans b_5 et b_6 . De même, la variable de plongement géométrique x_C est recopiée dans b_7 et b_8 . En particulier, tous les brins d'un même sommet topologique portent le même plongement géométrique.

Ainsi, la contrainte de plongement est satisfaite dans la partie droite de la règle.

À gauche de la Fig. 4.5(a), nous avons ajouté une précondition de plongement $proches(x_B, x_{A'}) \wedge proches(x_C, x_{D'})$ qui assure que la connexion s'effectue seulement si les sommets B et A' – de même que C et D' – sont suffisamment proches l'un de l'autre. Intuitivement, les préconditions de plongement sont des formules construites sur des expressions de plongement. Dans notre cas, *proches* est un prédicat qui est satisfait lorsque la distance qui existe entre les deux couples de coordonnées 2D passés en paramètre est en dessous d'un seuil fixé. Notons que le langage d'expression des préconditions de plongement est d'autant plus riche que ses primitives sont nombreuses. Notamment, nos préconditions – qui porte sur des distances – se composent facilement avec l'opérateur \wedge . D'une manière générale, les opérateurs logiques classiques ainsi que les opérateurs de comparaison sont essentiels. D'un point de vue pratique, remarquons que l'évaluation de la précondition de plongement nécessite le filtrage préalable de la partie gauche de la règle. Dans notre cas, ce filtrage permet l'instanciation des variables $x_B, x_C, x_{A'}, x_{D'}, p$ et p' .

4.3.3 Contrainte de plongement

En section 4.2, nous avons introduit la contrainte de plongement des cartes généralisées plongées. Cette contrainte nous indique que pour toute orbite plongée de type $\langle \beta \rangle$, deux brins qui appartiennent

à la même orbite $\langle \beta \rangle$ sont munis d'un même plongement d'orbite $\langle \beta \rangle$. Dans notre cas, les orbites plongées sont les orbites de type $\langle \alpha_0 \alpha_1 \rangle$ et $\langle \alpha_1 \alpha_2 \rangle$. Ainsi, deux brins qui appartiennent à la même orbite de type $\langle \alpha_1 \alpha_2 \rangle$ (respectivement $\langle \alpha_0 \alpha_1 \rangle$) possèdent le même plongement géométrique (respectivement biochimique). À la manière des contraintes d'arcs incidents et de cycles, que nous avons étendues aux règles de transformation, la contrainte de plongement doit être étendue aux règles qui manipulent les plongements.

Tout d'abord, dans une règle de transformation de G-cartes plongées, la contrainte de plongement des cartes généralisées plongées doit être satisfaite. Ainsi, dans la partie droite de la règle de connexion (voir Fig. 4.5(a)), les brins b_1 , b_2 , b_5 et b_6 – qui appartiennent à la même orbite sommet de type $\langle \alpha_0 \alpha_1 \rangle$ – sont munis d'un même plongement géométrique.

De plus, dans une règle de transformation de G-cartes plongées, il n'est pas permis de modifier le plongement des orbites incomplètes. Prenons l'exemple de la règle de connexion (voir Fig. 4.5(a)). Dans la partie droite de la règle, l'orbite $\langle \alpha_0 \alpha_1 \rangle$ (b_1) est incomplète car, notamment, les voisins de b_1 et de b_4 par α_0 ne sont pas filtrés par la règle. Nous ne pouvons donc pas modifier le plongement biochimique des brins b_1 à b_4 . Ainsi, on remarque que dans les parties gauche et droite de la règle de connexion, les brins b_1 à b_4 présentent la même variable de plongement biochimique p .

Sous couvert de ces deux contraintes ainsi que des contraintes d'arcs incidents et de cycle, nous sommes assurés qu'une règle préserve la cohérence des G-cartes plongées. Remarquons que les deux contraintes que nous venons d'explicitier s'étendent facilement aux règles avec renommages.

4.4 De l'utilité de manipuler plusieurs variables orbites

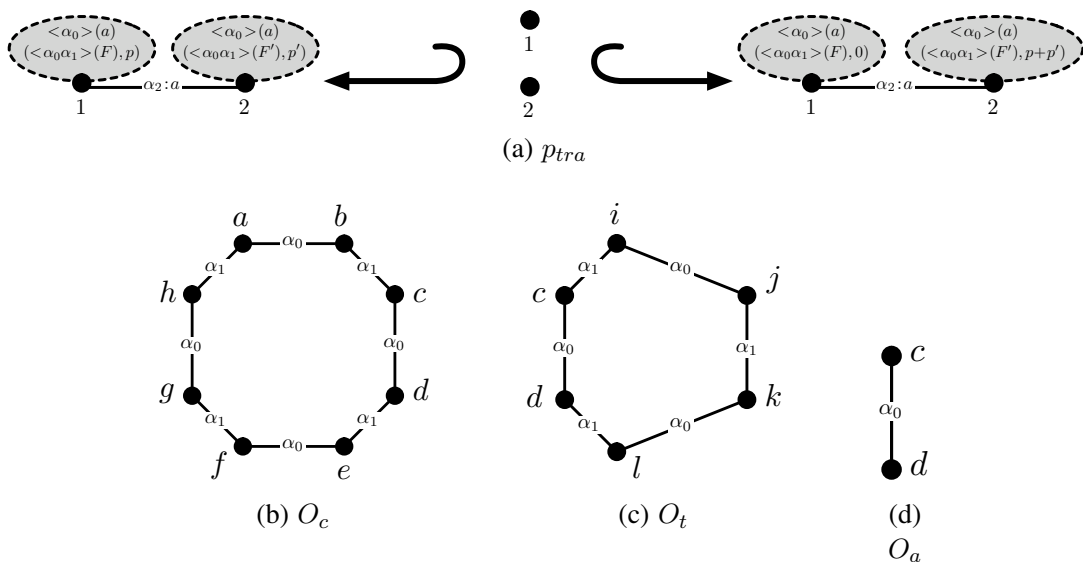


FIGURE 4.6 – La règle de transport p_{tra} et trois orbites

Dans le chapitre précédent, nous avons introduit des règles de transformation avec renommages d'orbites. Rappelons que ces règles sont munies d'une unique variable orbite. Lors de l'instanciation, les occurrences de la variable orbite sont remplacées par des orbites isomorphes au renommage près de leurs liaisons. Dans cette section, nous montrons l'intérêt de manipuler plusieurs variables orbites au sein d'une seule règle. Pour cela, nous prenons l'exemple de l'étape de transport du mécanisme des jonctions communicantes (voir Fig. 4.1(c)). Rappelons que lors de cette étape, deux cellules biologiques collées

l'une à l'autre s'échangent des protéines le long de leur surface de contact.

Nous proposons de traduire l'étape de transport par la règle $p_{tra} : L_{tra} \leftrightarrow K_{tra} \hookrightarrow R_{tra}$ illustrée en Fig. 4.6(a). Nous l'appelons la *règle de transport*. La règle p_{tra} est munie de trois variables orbites : deux variables F et F' de type $\langle \alpha_0 \alpha_1 \rangle$ et une variable a de type $\langle \alpha_0 \rangle$. Les variables F et F' abstraient les deux cellules biologiques représentées par des faces topologiques tandis que a abstrait la portion de membrane connectée des deux cellules biologiques. Ainsi, la partie gauche de la règle de transport modélise intuitivement deux faces F et F' qui portent respectivement les plongements biochimiques p et p' , et contiennent chacune une copie de l'arête a reliées deux à deux par α_2 .

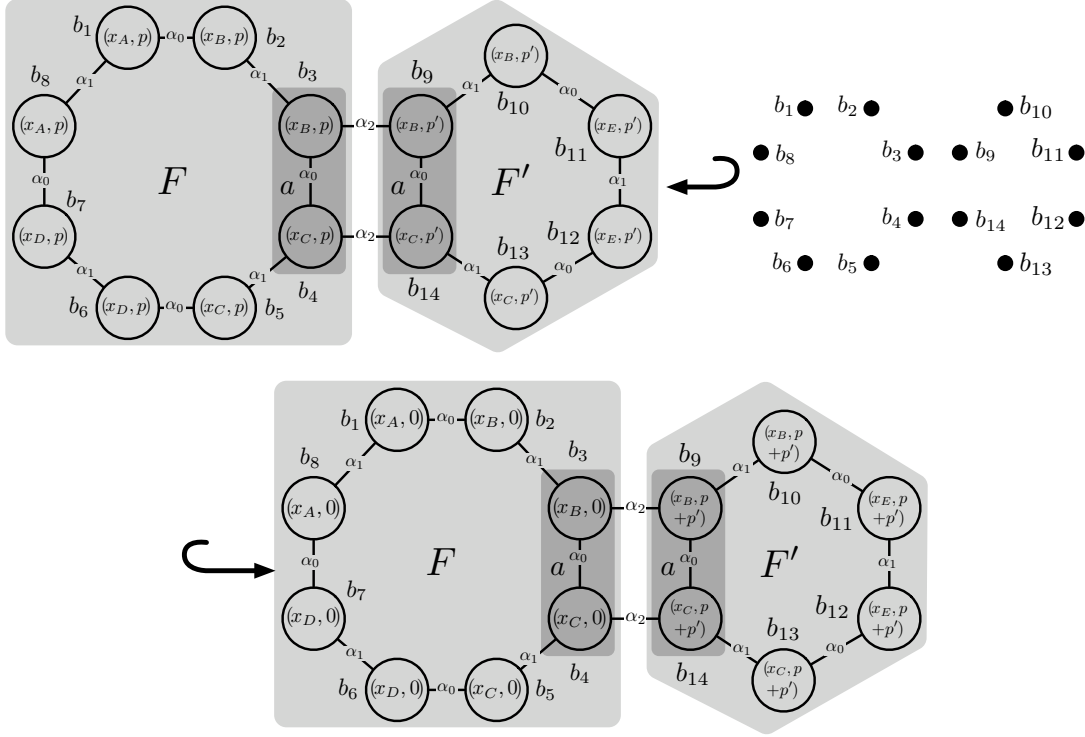


FIGURE 4.7 – p_{tra}^σ

La production p_{tra}^σ , résultat de l'instanciation de p_{tra} par une valuation σ illustrée en Fig. 4.7. La valuation σ instancie F par l'orbite carrée O_c (voir Fig. 4.6(b)), F' par l'orbite triangulaire O_t (voir Fig. 4.6(c)) et a par l'orbite O_a (voir Fig. 4.6(d)). Cette règle filtre deux cellules biologiques – l'une est de forme carrée et l'autre de forme triangulaire – et transporte les protéines de la cellule biologique carrée vers la cellule biologique triangulaire. La partie gauche de la règle de transport associée à l'orbite $\langle \alpha_0 \alpha_1 \rangle$ (F) une variable de plongement biochimique p . Ainsi, le sommet 1 abstrait la face topologique $\langle \alpha_0 \alpha_1 \rangle$ (b_1) de L_{tra}^σ (représentée par le carré grisé marqué par F sur la figure) et chaque sommet de F est plongé biochimiquement par la variable p . De la même manière, dans l'étiquette du sommet 2 figure l'orbite $\langle \alpha_0 \alpha_1 \rangle$ (F') muni d'une variable de plongement biochimique p' . Le sommet 2 abstrait alors la face triangulaire $\langle \alpha_0 \alpha_1 \rangle$ (b_9) (représentée par l'hexagone grisé marqué par F' sur la figure) et chaque sommet de la face est plongé biochimiquement par la variable p' .

Remarquons que dans la partie gauche de la règle de transport, parmi les étiquettes des sommets 1 et 2 figure également l'expression $\langle \alpha_0 \rangle$ (a). Ainsi, les sommets 1 et 2 abstraient également deux copies de la même arête topologique. De plus, les sommets 1 et 2 sont connectés par une arête étiquetée par α_2 : a . Intuitivement, cette étiquette signifie que les brins des deux copies $\langle \alpha_0 \rangle$ (a) abstraient par les sommets 1 et 2 sont connectés deux à deux par des liaisons α_2 . Ainsi, dans L_{tra}^σ , on remarque que

les brins des deux régions grisées marquées par a – il s’agit des orbites $\langle \alpha_0 \rangle$ abstraient par 1 et 2 – sont reliés par α_2 . Notons que l’étiquette d’un arc $\alpha_2 : a$ n’a de sens que si les sommets connectés par cet arc présentent la variable a dans leur étiquette. Enfin, dans la partie droite p_{tra}^σ , on remarque que les brins de la face F' présentent tous l’expression de plongement biochimique $p + p'$ tandis que ceux de la face F présentent le plongement biochimique 0. Cette réécriture du plongement biochimique modélise le transport des protéines depuis la cellule biologique modélisée par F vers la cellule biologique modélisée par F' .

Ainsi, dans cet exemple, l’utilisation de deux variables distinctes F et F' de type $\langle \alpha_0 \alpha_1 \rangle$ nous permet de filtrer deux faces topologiques quelconques de formes différentes. De plus l’utilisation d’une troisième variable a de type $\langle \alpha_0 \rangle$ nous permet de modéliser la connexion par α_2 de deux arêtes topologiques isomorphes.

Remarquons toutefois que la mise en jeu de plusieurs variables orbites au sein de la règle de transport ne nous permet pas d’utiliser directement les mécanismes d’instanciation par produits fibrés introduits dans la chapitre 3. Nous devons donc étendre nos travaux afin de généraliser l’instanciation des règles aux cas de plusieurs variables. Une piste possible consiste à traiter indépendamment les différentes variables, puis à recoller entre eux les graphes résultats de l’instanciation de chaque variable.

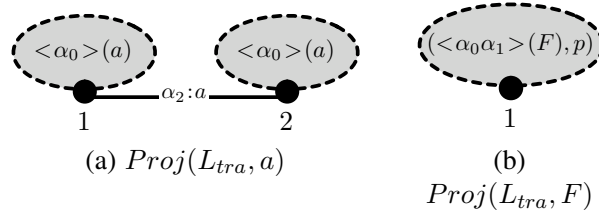
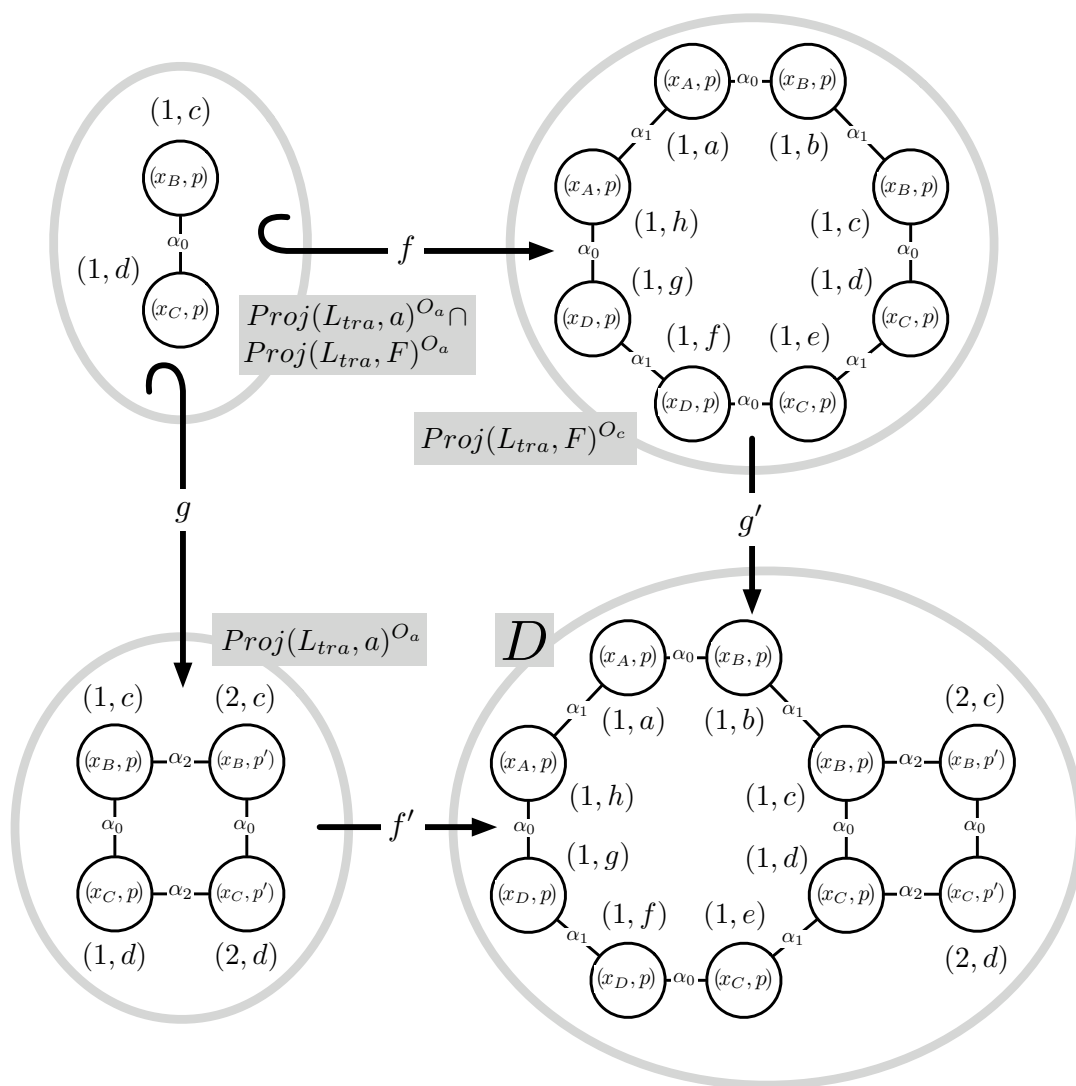


FIGURE 4.8 – Projection de L_{tra} sur les variables a et F

Illustrons ce principe dans le cas de l’instanciation de la partie gauche de la règle de transport. Afin de rendre l’exemple plus pédagogique, nous manipulons uniquement les variables a et F . En Fig. 4.8, nous projetons indépendamment L_{tra} sur les variables a et F . La projection $Proj(L_{tra}, a)$ de L_{tra} sur la variable a est illustrée en Fig. 4.8(a). Intuitivement, la projection sur a nous permet d’isoler le sous-graphe de L_{tra} étiqueté par la variable a . Ainsi, le graphe $Proj(L_{tra}, a)$ ne comprend plus qu’une unique variable a . Nous pourrions donc l’instancier selon les mécanismes décrits dans le cas de la règle de synthèse (voir section 4.3.1) qui associe l’instanciation des renommages d’orbite (voir section 3.3.5 page 77) à celle des variables attributs. De la même manière, la projection $Proj(L_{tra}, F)$ de L_{tra} sur la variable F est illustrée en Fig. 4.8(b). Ici encore, nous pouvons instancier $Proj(L_{tra}, F)$ à l’aide du mécanisme d’instanciation des règles de modification de plongement.

L’instanciation $Proj(L_{tra}, a)^{O_a}$ de $Proj(L_{tra}, a)$ par l’orbite $\sigma(a) = O_a$ est illustrée en bas à gauche du diagramme de la Fig. 4.9. L’instanciation $Proj(L_{tra}, F)^{O_c}$ de $Proj(L_{tra}, F)$ par l’orbite $\sigma(F) = O_c$ est illustrée en haut à droite du diagramme de la Fig. 4.9. Nous souhaitons à présent recoller les graphes $Proj(L_{tra}, a)^{O_a}$ et $Proj(L_{tra}, F)^{O_c}$ par la somme amalgamée le long de leur intersection $Proj(L_{tra}, a)^{O_a} \cap Proj(L_{tra}, F)^{O_c}$. Le calcul de l’intersection est donné immédiatement par les intersections des sous-graphes $Proj(L_{tra}, a)^{O_a}$ et $Proj(L_{tra}, F)^{O_c}$. De la même manière, les morphismes f et g sont donnés par les inclusions de $Proj(L_{tra}, a)^{O_a} \cap Proj(L_{tra}, F)^{O_c}$ dans respectivement $Proj(L_{tra}, a)^{O_a}$ et $Proj(L_{tra}, F)^{O_c}$. Enfin, le graphe D résultat du collage par somme amalgamée est illustré en bas à droite du diagramme. Remarquons que le graphe D correspond bien au sous-graphe comprenant les régions grisées F et a dans la partie gauche de l’instanciation p_{tra}^σ (voir Fig. 4.7). Par une construction analogue, nous pourrions construire l’intégralité de la partie gauche de p_{tra}^σ en projetant L_{tra} sur la variable F' et en recollant son instanciation par O_t au graphe D le long de leur intersection.

FIGURE 4.9 – Collage de $Proj(L_{tra}, a)$ et $Proj(L_{tra}, F)$

Deuxième partie

Application à l'étude de la dynamique de l'appareil de Golgi

L'APPAREIL DE GOLGI

Sommaire

5.1	La voie sécrétoire	120
5.2	Trois modèles pour l'appareil de Golgi	123
5.2.1	Modèle vésiculaire	123
5.2.2	Modèle de la maturation	126
5.2.3	Modèle du flux continu de membranes	128
5.3	Bilan	131

Dans ce chapitre, nous présentons notre cas d'étude biologique : l'*appareil de Golgi*. Cette organelle, qui prend part à la voie sécrétoire de la cellule biologique, présente une structure méconnue où la notion de compartiment apparaît fondamentale. À ce titre, les modèles géométriques à base topologique tels que les cartes généralisées semblent tout indiqués de par leurs relations de voisinage qui permettent de modéliser l'agencement des compartiments. Dans les chapitres suivants, nous nous appuyons sur cette étude de l'appareil de Golgi afin d'introduire les outils que nous développons en partenariat avec nos collègues biologiques.

La voie sécrétoire de la cellule biologique est le chemin suivi par les protéines depuis leur excrétion du noyau jusqu'à l'environnement externe à la cellule. Il est admis que pour l'essentiel, le transport des protéines est le fait de mécanismes d'empaquetage et de dépaquetage dans les compartiments intracellulaires qui longent la voie sécrétoire. Parmi l'ensemble de ces compartiments, l'appareil de Golgi joue un rôle fédérateur. Il est une véritable tour de contrôle qui trie et guide les protéines vers la membrane plasmique qui borde la cellule biologique.

Cependant, il est à noter que si la fonction du Golgi semble aujourd'hui bien connue, sa structure dans l'espace reste sujette à débats pour l'ensemble de la communauté. En effet, comme la plupart des objets biologiques qui constituent la cellule, l'appareil de Golgi ne peut être observé qu'au prix de contraintes fortes. Deux mécanismes d'observation existent : la microscopie optique, qui conjugue dynamique (les objets peuvent être observés vivants) et faible résolution des images tandis que la microscopie électronique offre de meilleures résolutions mais uniquement sur de fines coupes inertes. Or, d'après les biologistes spécialistes de l'appareil de Golgi, il apparaît que la structure spatiale du Golgi soit inextricablement liée à sa dynamique. Par conséquent, de nombreux modèles – ou hypothèses – « spatio-temporelles » de l'appareil Golgi existent sans que l'une ou l'autre ne puisse s'imposer.

Dans ce chapitre, nous évoquons quelques généralités sur la cellule biologique puis introduisons la voie sécrétoire et enfin l'appareil de Golgi (voir section 5.1). Nous présentons également trois des principaux modèles de l'appareil de Golgi, aujourd'hui encore en concurrence (voir respectivement les sections 5.2.1, 5.2.2 et 5.2.2). Une grande part des connaissances biologiques évoquées dans ce chapitre, notamment au sujet des trois modèles, est issue de [Képès 06]. Cet article s'appuie notamment sur certains travaux [Farquhar 81, Mironov 97, Rambourg 79, Mollenhauer 91] afin de comparer les mécanismes essentiels à la base de ces trois modèles de l'appareil de Golgi.

Avant tout, notons que puisque le présent chapitre traite dans une large mesure de cellules biologiques, nous utilisons le terme « cellule » dans son sens biologique. Sauf mention contraire, nous oublions son sens topologique introduit dans la première partie du manuscrit (voir section 1.3.2).

5.1 La voie sécrétoire

On distingue deux types de cellules biologiques : les cellules procaryotes et les cellules eucaryotes. Les procaryotes sont environ 20 fois plus petites que les eucaryotes (si on se base sur leur rayon moyen), ce sont les cellules des bactéries. Les procaryotes peuvent être de diverses formes : de la capsule au bâtonnet. Le cytoplasme des procaryotes, c'est-à-dire leur région interne, est enveloppé d'une paroi externe rigide qui donne sa forme à la cellule. La paroi externe permet également aux procaryotes de résister à l'éclatement face à la forte pression – on parle de « pression osmotique » – qui règne dans le cytoplasme. Notons que sans enveloppe rigide, la pression osmotique induirait une forme sphérique aux cellules procaryotes. La distinction fondamentale à noter entre les procaryotes et les eucaryotes est l'absence – chez les procaryotes – de noyau, qui chez les eucaryotes enferme le matériel génétique de la cellule (c'est-à-dire la double hélice d'ADN et donc les gènes).

Le rapport de surface – nous parlons ici de la surface membranaire externe – entre les cellules eucaryotes et procaryotes est de $400 = 20^2$, leur rapport de volume est de $8000 = 20^3$. Nous savons que l'activité métabolique de la cellule, intuitivement son activité interne, est proportionnelle au volume tandis que les échanges de molécules sont proportionnels à la surface (également appelée surface d'échange). Il apparaît donc que comparées aux cellules procaryotes, les cellules eucaryotes présentent un déficit de $20^3/20^2 = 20$ dans le rapport de leur activité métabolique à leur surface d'échange. Or, toute activité métabolique dépend des échanges de matière permis par les surfaces membranaires. Afin de combler ce fort déficit en membranes, les cellules eucaryotes – qui sont les cellules des plantes et des animaux – présentent une grande quantité de membranes repliées dans leur cytoplasme, on parle d'endomembrane. Cette abondance d'endomembranes dans le cytoplasme induit une forte compartimentation de la cellule eucaryote. Dans ce chapitre, nous nous intéressons à certains de ces compartiments qui prennent part à la voie sécrétoire de la cellule eucaryote.

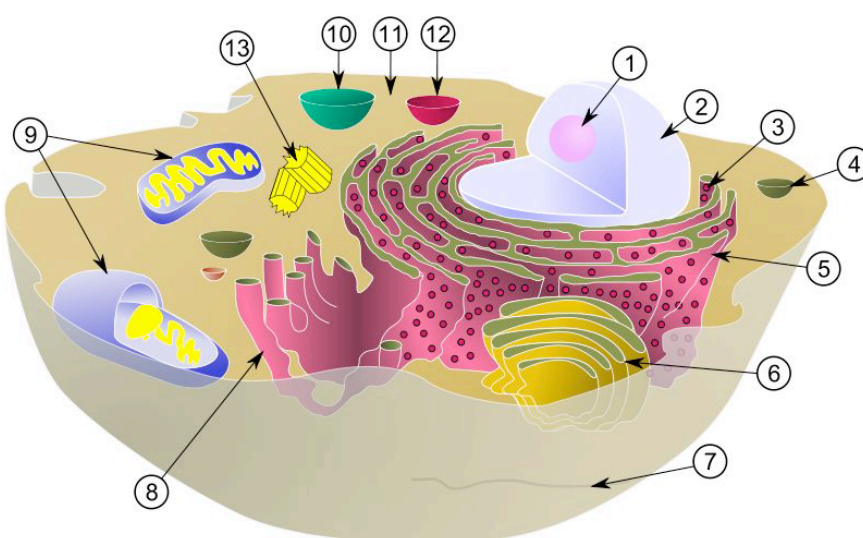


FIGURE 5.1 – Une cellule eucaryote. Le nucléole (1), le noyau (2), un ribosome (3), une vésicule (4), le réticulum endoplasmique rugueux (ou granuleux) (5), l'appareil de Golgi (6), un microtubule (7), le réticulum endoplasmique lisse (8), deux mitochondries (9), la vacuole (10), le cytoplasme (11), un lysosome (12), le centrosome (13). Source : [Wik]

Une représentation schématique d'une cellule eucaryote typique est proposée en Fig. 5.1. Les organites, ou organelles, sont les petites structures qui habitent la cellule. Certains d'entre-eux, parmi les plus connus, sont annotés sur la figure. Intéressons-nous aux organites de la voie sécrétoire en reconstituant le cheminement des protéines depuis leur synthèse à la périphérie du noyau, jusqu'à leur passage à l'extérieur de la cellule.

Toute protéine synthétisée dans la cellule est issue d'un gène. Intuitivement, un gène est une petite portion d'ADN contenue dans le noyau (voir l'annotation 2 sur la figure). Remarquons que contrairement à certaines idées reçues, le noyau n'est pas systématiquement situé au centre de la cellule. Bien que cela n'apparaisse pas sur la figure, il n'est pas non plus systématiquement de forme arrondie. Le passage du gène à la protéine nécessite d'effectuer une copie du gène dans une nouvelle molécule : l'ARN. Ce mécanisme de copie est la transcription, elle a lieu dans le nucléole du noyau (voir 1).

L'ARN traverse la membrane du noyau, également appelée enveloppe nucléaire, jusqu'au cytoplasme de la cellule (voir l'annotation 11 sur la figure). Les ribosomes (voir 3), situés généralement à la périphérie du noyau, participent à la traduction de l'ARN en séquences d'acides aminés : les protéines. Notons

que le cytoplasme n'est pas une simple poche de liquide dans laquelle les organites sont en suspension. En réalité, il est un environnement extrêmement visqueux et à l'échelle de la cellule, même les molécules d'eau présentes dans le cytoplasme sont un obstacle au déplacement des protéines. Pour cette raison, les mécanismes de diffusion¹, qui ont naturellement tendance à transporter les protéines depuis les régions où elles sont le plus concentrées (autour du noyau) vers les régions où elles le sont moins, ne suffisent pas à expliquer les cinétiques observées par les biologistes. Ainsi, le chemin suivi par les protéines est constitué de structures dites accélératrices qui elles seules assurent des cinétiques convenables pour la cellule.

La première de ces structures est le *réticulum endoplasmique* (ou *RE*) qui est un compartiment du cytoplasme et qui entoure le noyau de la cellule. Il est constitué de citernes aplaties et de tubes délimités par des membranes. On distingue deux portions du réticulum endoplasmique : une partie rugueuse (ou granuleuse, voir l'annotation 5 sur la figure), où de nombreux ribosomes sont accolés à la membrane, et une partie lisse (voir 8) dépourvue de ribosome. Les protéines, synthétisées dans la région rugueuse, sont véhiculées le long du RE puis transportées jusqu'à l'*appareil de Golgi* (voir 6) par l'intermédiaire de petites vésicules (voir 4). Intuitivement, les vésicules sont de petits compartiments de forme sphérique. Des mécanismes d'emballage et de dépaquetage des protéines dans les vésicules permettent d'assurer leur transport d'un compartiment à un autre, notamment du RE à l'appareil de Golgi. Il est à noter que si la présence de vésicules dans la cellule biologique semble être acceptée par l'ensemble des biologistes, leur localisation fine et donc leur rôle dans telle ou telle région de la cellule n'est pas encore arrêté. Ces vésicules sont des objets cellulaires où les limites de l'observation ne permettent de conclure et nous verrons dans ce chapitre qu'elles jouent un rôle important dans les différents modèles de l'appareil de Golgi.

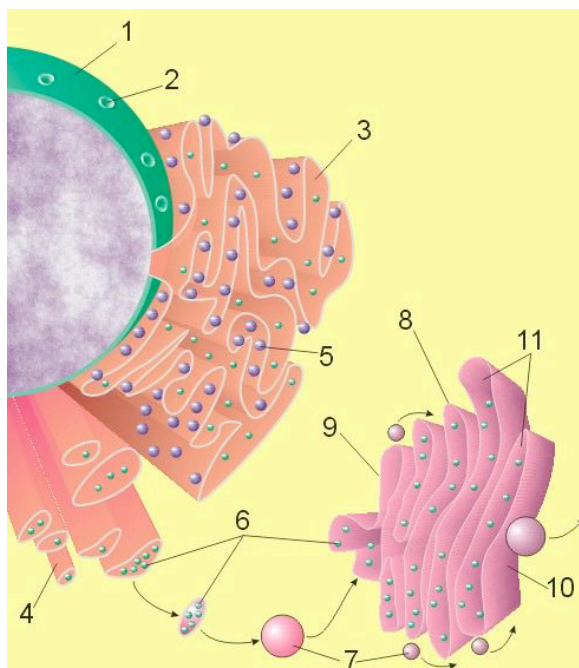


FIGURE 5.2 – La voie sécrétoire chez l'animal. Le noyau (1), un pore nucléaire (2), le réticulum endoplasmique rugueux (3), le réticulum endoplasmique lisse (4), un ribosome (5), des protéines transportées (6), des vésicules (7), l'appareil de Golgi (8), la face *cis* (9), la face *trans* (10), deux saccules (11). Source : [Wik]

1. La diffusion désigne la tendance naturelle d'un système à rendre homogènes les concentrations des espèces chimiques en son sein.

L'appareil de Golgi a été mis en évidence en 1898 par Camillo Golgi [Golgi 98] alors qu'il observait un arrangement particulier du RE. Sa découverte, qu'il nomma « appareil réticulaire interne », fut dès l'origine sujet à controverses en raison de la méthode d'observation utilisée. Depuis, l'appareil fut renommé en « appareil de Golgi » en raison de l'importance, *a posteriori*, de sa découverte. L'appareil de Golgi est l'un des organites fondateurs de la voie sécrétoire. Sa principale contribution est le transport des protéines depuis le RE vers différentes régions de la cellule : par exemple la membrane plasmique ou les lysosomes (voir l'annotation 12 sur la figure). En plus d'une tour de contrôle chargée de trier puis de diriger les protéines en fonction de leur type biologique, l'appareil de Golgi est aussi le lieu de la maturation des protéines. Le processus de maturation est assuré par la perte de séquences peptidiques² puis par l'ajout de sucres (on parle alors de glycosylation) ou de sulfates (sulfatation). La structure de l'appareil de Golgi peut varier d'une espèce à une autre. Chez la plante, il est constitué de plusieurs centaines d'éléments distincts, appelés dictyosomes. Il est généralement unique chez les animaux. Dans la suite du chapitre, nous nous plaçons dans le cadre de la cellule animale.

Dans la littérature, l'appareil de Golgi est généralement représenté tel qu'en Fig. 5.2. Commentons brièvement cette représentation. On remarque que l'appareil de Golgi apparaît comme constitué de 5 ou 6 citernes déconnectées : les saccules (voir annotation 11 sur la figure). Parmi les saccules, on distingue : la face *cis* (voir annotation 9), orientée vers le RE, et la face *trans* (voir annotation 10), souvent orientée vers la membrane plasmique. Ainsi, on dit que l'appareil de Golgi est une structure polarisée de *cis* vers *trans*. Enfin, l'appareil de Golgi apparaît généralement bordé par de petites vésicules (voir annotation 7).

5.2 Trois modèles pour l'appareil de Golgi

Comme nous l'avons vu en introduction, l'observation par microscopie optique et électronique ne permet pas de conclure quant à la forme précise et à la dynamique de l'appareil de Golgi. Ainsi, bien que la représentation donnée en Fig. 5.2 soit la plus communément admise, elle n'en reste pas moins un modèle qui, par définition, abstrait et altère la réalité de l'appareil de Golgi.

Dans cette section, nous introduisons trois modèles de fonctionnement de l'appareil de Golgi. Dans ces modèles, plus précis que la représentation proposée dans la section précédente, nous nous intéressons à la structure ainsi qu'à la dynamique de l'appareil. En particulier, nous verrons de quelle manière la structure peut influencer fortement sur les aspects dynamiques.

Le premier modèle, le *modèle vésiculaire* (section 5.2.1), est le plus proche de la représentation proposée en Fig. 5.2. Le second modèle, le *modèle de la maturation* (section 5.2.2), se distingue par une autre approche des aspects dynamiques. Enfin, le *modèle du flux continu de membranes* (section 5.2.3), aborde sous un autre angle les connaissances tant structurelles que dynamiques.

5.2.1 Modèle vésiculaire

Sur de nombreuses images issues de la microscopie électronique, l'appareil de Golgi apparaît comme étant constitué d'une pile de saccules déconnectés. Sur le micrographe de la Fig. 5.3, où l'on observe une fine coupe d'appareil de Golgi d'une cellule endothéliale³ de souris, on remarque cinq saccules déconnectés dans la région S. Cette observation corrobore l'hypothèse d'un appareil de Golgi vésiculaire [Farquhar 81, Farquhar 85, Wattenbreg 91, Rothman 94, Robinson 98, Hawes 99], où le transport des protéines est expliqué par la présence de vésicules qui « sautent » d'un saccule à un autre. Sur le micrographe, de telles vésicules apparaissent à la périphérie de la pile, notamment dans la région V. On remarque enfin des grains de sécrétion aux abords de la face *trans* (voir région G sur la micrographie). Une représentation schématique du modèle vésiculaire est proposée en Fig. 5.4. Sur la figure, seuls trois

2. Une séquence peptidique est une petite chaîne d'acides aminés.

3. Les cellules endothéliales tapissent la paroi des vaisseaux sanguins et lymphatiques.

sacculs déconnectés sont représentés. Deux d'entre eux représentent les faces *cis* et *trans*. Les vésicules sont illustrées dans les régions V.

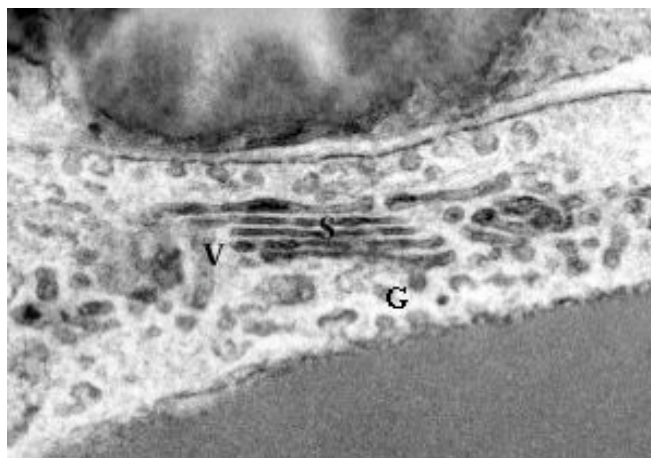


FIGURE 5.3 – Micrographie d'un appareil de Golgi d'une cellule endothéliale de souris, avec l'aimable autorisation de Jean-Marc Verbavatz (Institut IBITEC-S, CEA-Saclay)

Commentons les mécanismes dynamiques sous-jacents à ce modèle. La face *cis* reçoit de petits compartiments, appelées *précurseurs*, issues du RE (voir flèche 1 sur la Fig. 5.4). Les protéines de la voie sécrétoire sont transmises du RE vers l'appareil de Golgi par l'intermédiaire des précurseurs. Il est à noter que dans la littérature, il est parfois d'usage d'utiliser le terme « vésicule » pour nommer les précurseurs. Cependant, nous préférons réserver ce terme aux vésicules qui transportent les protéines entre les saccules. Remarquons enfin que l'on distingue deux types de protéines : les *protéines solubles* sont véhiculées dans la lumière – c'est-à-dire à l'intérieur – des compartiments, tandis que les *protéines membranaires* effectuent leur voyage sur les membranes. Lorsque nous utilisons le terme « protéine » sans le qualifier, nous faisons référence aux deux espèces simultanément.

Étudions à présent les mécanismes vésiculaires mis en jeu dans le transport des protéines. Ainsi, on suppose que des vésicules se déplacent d'un saccule donné vers son saccule supérieur en suivant un mouvement antérograde de la face *cis* vers la face *trans* (voir flèche 2 sur la Fig. 5.4). Ce transfert des vésicules explique le transport des protéines. Ainsi, les protéines membranaires viennent s'insérer dans la membrane des vésicules tandis que les solubles s'installent dans leur lumière. On parle de *bourgeonnement* pour désigner le mécanisme de création d'une vésicule à la surface d'un saccule. Les réactions biochimiques impliquées dans le bourgeonnement des vésicules sont relativement complexes et d'un faible intérêt dans le contexte de ce manuscrit. Cependant, nous pouvons en simplifier raisonnablement le principe en supposant qu'une espèce moléculaire, que nous appelons *activateur*, est à l'origine du bourgeonnement (voir Fig. 5.5). Les activateurs sont enchassés le long de la membrane périphérique des saccules, ils peuvent être dans deux états inter-dépendants. À l'état *inactif*, un activateur diffuse le long de la membrane périphérique (voir Fig. 5.5(a), l'activateur inactif est représenté par un point gris). Il peut à tout moment entrer dans l'état *actif*. Dès lors, l'activateur induit une courbure membranaire responsable de la création d'une vésicule (voir Fig. 5.5(b), l'activateur actif est représenté par un point noir). Suite au bourgeonnement, l'activateur entre dans l'état inactif et diffuse à nouveau (voir Fig. 5.5(c)).

Dans la section 5.1, nous avons vu que l'appareil de Golgi tient un double rôle dans la voie sécrétoire. En premier lieu, il assure le transport des protéines depuis le RE vers les régions proches de la membrane plasmique. En second lieu, il est le coordinateur de la maturation des protéines. Rappelons que, intuitivement, la maturation est un phénomène biochimique qui transforme (ou mature) les protéines. Or, on

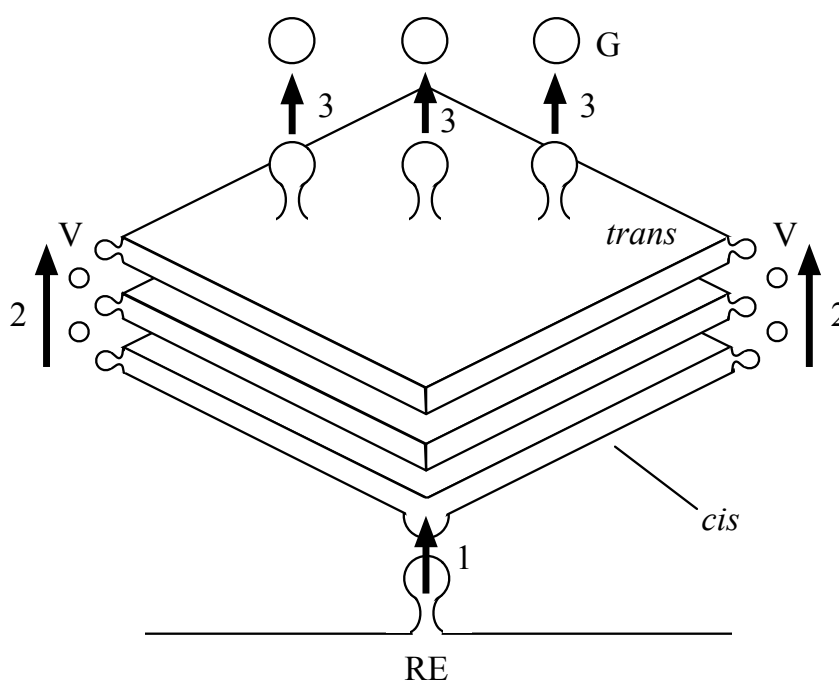


FIGURE 5.4 – Le modèle vésiculaire

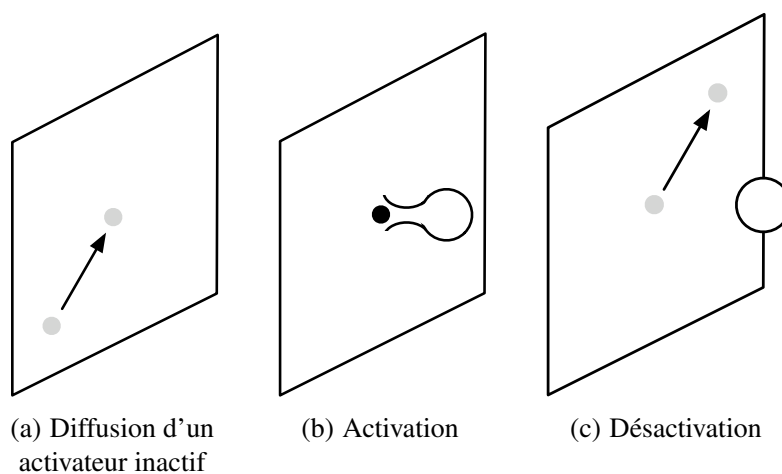


FIGURE 5.5 – Bourgeonnement d'une vésicule

sait que certaines enzymes⁴ sont responsables des réactions mises en jeu dans la maturation. De plus, des expériences ont montré que ces enzymes sont en grande partie localisées dans les saccules proches de la face *cis*. Dans notre cas, comme nous ne représentons que trois saccules, et sans que cela ne fausse notre discours, nous supposons que les enzymes sont situées dans la face *cis*. Dans le modèle vésiculaire, la pile de saccules est immobile. Ainsi, en supposant que les enzymes ne sont pas impliquées dans un transport vésiculaire, elles stationnent dans la face *cis* par définition.

Enfin, lorsque les protéines (sous-entendu maturées) atteignent la face *trans*, elle sont empaquetées dans des grains de sécrétion (voir G sur la Fig. 5.4) qui bourgeonnent depuis la membrane. Les grains de

4. Un (ou une) enzyme est un catalyseur biologique, c'est-à-dire une espèce moléculaire qui accélère les réactions chimiques.

sécrétion peuvent être vus comme de grosses vésicules assurant le transport des protéines depuis l'appareil de Golgi vers la membrane plasmique.

Avant de conclure cette présentation du modèle vésiculaire, rappelons qu'en règle générale, plusieurs espèces de protéines sont présentes dans l'appareil de Golgi à un même instant. Ces espèces ne se distinguent pas seulement selon qu'elles sont de nature soluble ou membranaire. En fait, il existe de nombreuses espèces de protéines solubles (respectivement membranaires). Ajoutons que des expériences ont montré que le voyage des protéines, depuis leur entrée par la face *cis* jusqu'à leur sortie par les grains de sécrétion, peut être plus ou moins long en fonction de l'espèce considérée. L'hypothèse d'un appareil de Golgi vésiculaire est compatible avec ces observations. En effet, on suppose que certaines espèces protéiques possèdent une affinité plus grande pour la région centrale des saccules, par rapport à leur région périphérique. Par conséquent, de telles protéines sont moins souvent séquestrées⁵ par les vésicules que les protéines d'affinité contraire, c'est-à-dire celles ayant le moins d'affinité avec la région centrale.

5.2.2 Modèle de la maturation

Le modèle de la maturation [Mironov 97, Mironov 98, Bonfanti 98] est illustrée en Fig. 5.6. D'un point de vue de la structure, la représentation de l'appareil de Golgi proposée dans ce second modèle est la même que pour le modèle vésiculaire (voir Fig. 5.4). C'est lorsque l'on étudie la dynamique que les deux modèles se distinguent. Rappelons que dans le modèle vésiculaire, la pile de saccules est fixe et le transport des protéines depuis la face *cis* vers la face *trans* est le fait d'un mouvement antérograde des vésicules. Dans le modèle de la maturation, les saccules migrent dans le sens antérograde (voir flèches 4 sur la figure), on dit qu'elles mûrissent, et leur mouvement est responsable du transport des protéines. Ainsi, dans l'expression « modèle de la maturation », le terme « maturation » désigne le mouvement des saccules et non les transformations que subissent les protéines lors de leur traversée de l'appareil de Golgi. Dans ce paragraphe, pour lever toute ambiguïté, nous utiliserons l'expression « maturation des protéines » lorsque nous évoquerons leur transformation par les enzymes.

Ajoutons qu'historiquement, l'introduction du modèle de la maturation a été motivée par le coût énergétique élevé du modèle vésiculaire. D'après les biologistes, les nombreux mécanismes d'emballage et de dépaquetage des vésicules dans le modèle vésiculaire sont coûteux en énergie. Ainsi, dans le modèle vésiculaire, le transport des vésicules le long de l'appareil de Golgi paraît inefficace. Des biologistes ont alors émis l'hypothèse d'un transport des protéines *via* un mouvement antérograde des saccules, *a priori* plus économe.

Le mouvement antérograde des saccules est motivé par deux hypothèses. Tout d'abord, rappelons que sur les images issues de la microscopie électronique, tous les objets visibles sont immobiles par définition. En effet, en dépit des bonnes résolutions proposées par les microscopes électroniques, les images sont toujours prises sur de fines coupes inertes. Ainsi, on peut toujours supposer qu'*in vivo* certains des objets observés sont en mouvement. Par exemple, sur le micrographe de la Fig. 5.3, on peut supposer que les saccules (dénnotés par S sur la figure) sont mobiles.

La seconde hypothèse en faveur d'un mouvement antérograde des saccules est liée au bourgeonnement des grains de sécrétion. On peut supposer que le bourgeonnement des grains de sécrétion (voir flèches 3 sur la Fig. 5.6) induise un désagrégement complet de la face *trans*. Dans ce cas, la disparition de la face *trans* doit être compensée par la création d'un nouveau saccule *cis*, par fusion des précurseurs issus du RE (voir flèche 1). On suppose alors que le nouveau saccule agit mécaniquement sur la pile en poussant les saccules dans le sens antérograde. Ainsi, le mécanisme de renouvellement des saccules serait à l'origine de la maturation de la pile. Une conséquence de la maturation est que chaque saccule

5. En biologie, le terme « séquestré » est souvent utilisé pour qualifier une zone de forte concentration en une molécule donnée.

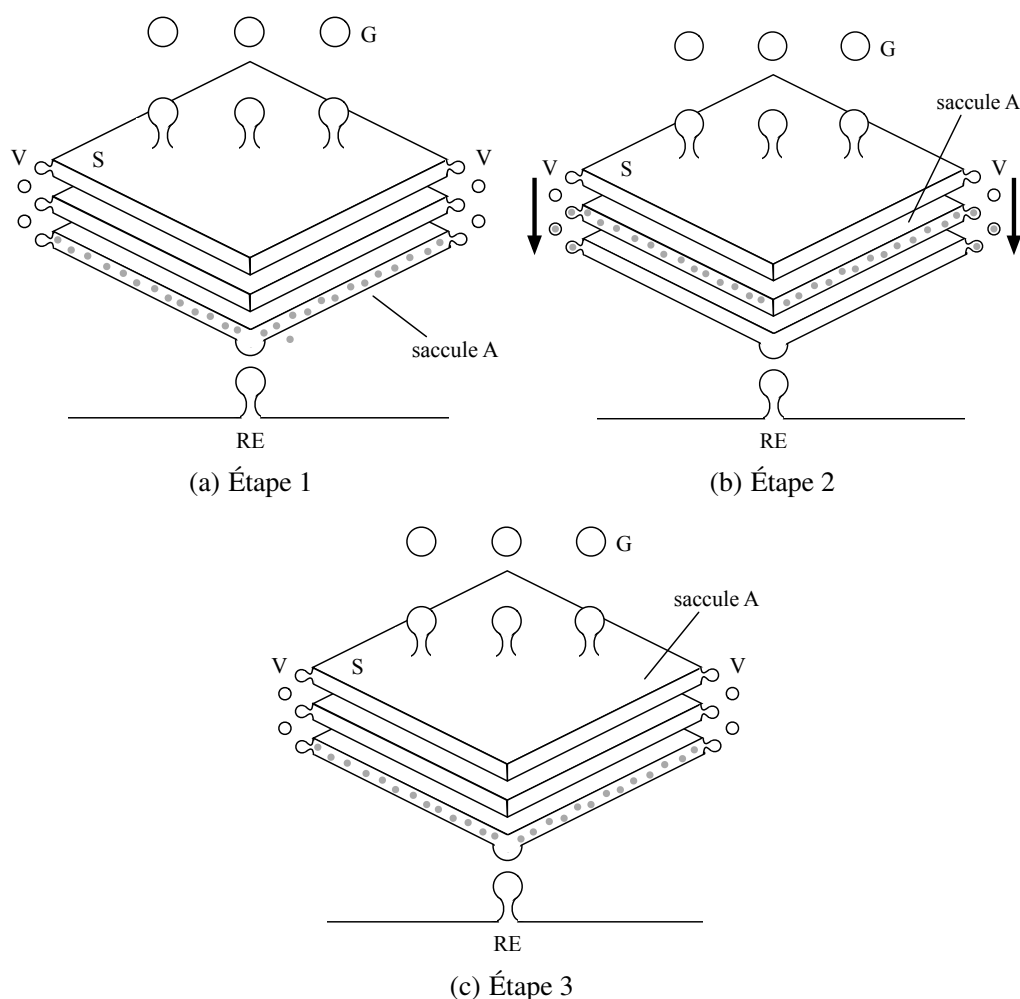


FIGURE 5.7 – Transport rétrograde des enzymes

5.2.3 Modèle du flux continu de membranes

Le modèle du flux continu de membranes se distingue du modèle vésiculaire et du modèle de la maturation tant sur le plan structural que sur le plan dynamique. Nous proposons une représentation de ce nouveau modèle en Fig. 5.8. On remarque que les vésicules ont disparu, elles sont remplacées par des *tubes* qui connectent les saccules (voir régions T sur la figure).

Bien que sur la plupart des micrographes de l'appareil de Golgi (voir Fig. 5.3) les tubes n'apparaissent pas en tant que tels, ce qui est usuellement interprété comme une fine section de vésicule peut également être interprété comme une petite section de tube. Illustrons cette remarque à l'aide de la Fig. 5.9. Sur la Fig. 5.9(a), une sphère – qui modélise une vésicule – est coupée par un plan. Le résultat de la coupe est illustré en Fig. 5.9(b). Une coupe analogue est effectuée sur un cylindre – qui modélise un tube – en Fig. 5.9(c) et 5.9(d). Dans l'exemple, on remarque que les résultats des coupes sont identiques. Cette étude simple met en exergue l'ambiguïté inhérente à l'observation d'objets 3D par des coupes. D'une manière générale, lorsque l'on observe des objets biologiques en microscopie, optique ou électronique, il est souvent difficile de conclure sur la structure réelle du sujet. L'appareil de Golgi notamment, est un cas d'étude où les biologistes sont confrontés aux limites des techniques d'observation. Ainsi, même si dès 1960, des expériences ont montré la présence de zones tubulaires à la périphérie de l'appareil de Golgi (voir [Mollenhauer 91] publié en 1991), seules des techniques plus récentes ont permis de confirmer

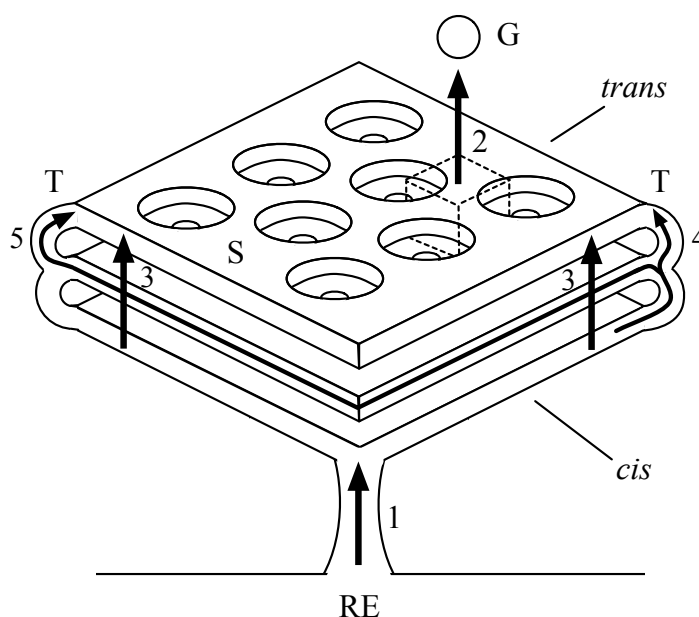


FIGURE 5.8 – Le modèle du flux continu de membranes

mer ces observations.

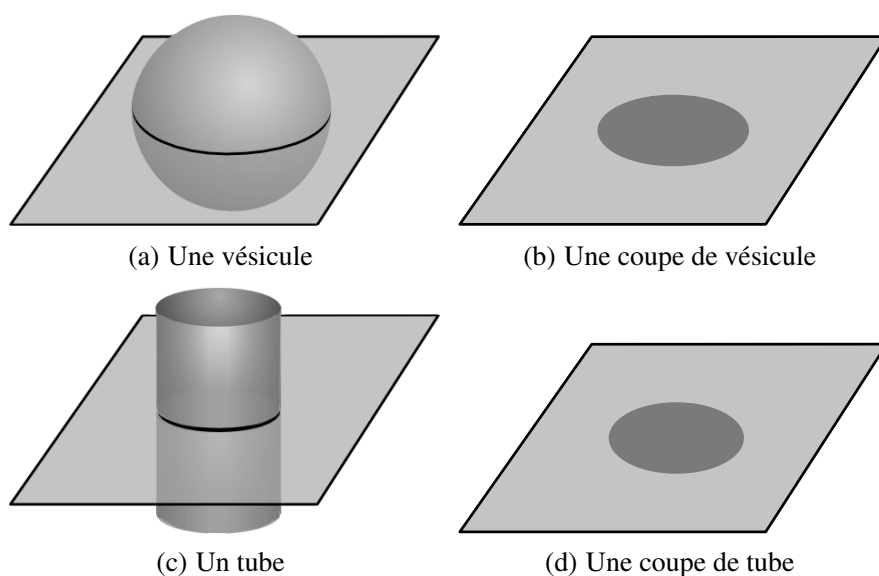


FIGURE 5.9 – Ambiguïté de l'observation par coupes

Parmi ces nouvelles techniques, la vision stéréoscopique [Rambourg 69, Rambourg 79] et l'animation 3D [Képès 03] permettent d'observer les zones tubulaires de manière non équivoque. Dans les deux cas, il s'agit d'observer l'appareil de Golgi sur des coupes épaisses. En vision stéréoscopique, deux images de la coupe sont prises sous deux angles différents. L'assemblage des deux images permet alors d'observer l'appareil de Golgi en 3 dimensions. En « animation 3D », il s'agit de réaliser une courte séquence vidéo à partir de plusieurs images prises sous différents angles. La lecture de l'animation donne un effet de relief qui permet de mieux interpréter la structure des objets.

Dans le modèle du flux continu de membranes, on suppose que les vésicules observées sont de petites sections de tubes qui connectent les saccules. Du fait de cette hypothèse d'un appareil de Golgi continu, le transport des protéines depuis la région *cis* vers la région *trans* est effectué au sein d'un unique compartiment. Ajoutons que comme dans le modèle de la maturation, l'ensemble de la structure migre dans le sens antérograde (voir flèches 3 sur la Fig. 5.8). Ainsi, le mouvement des protéines est à la fois induit par le mouvement de la structure et par leur diffusion le long des saccules et des tubes.

De cette manière, plusieurs espèces de protéines peuvent être transportées selon des cinétiques différentes. Ce modèle paraît recevable puisque l'on sait, notamment, que les protéines d'une même espèce sont de même taille, et que cette taille influe sur la vitesse de diffusion. De plus, nous avons vu que les espèces protéiques peuvent avoir des affinités particulières pour certaines régions du Golgi (voir section 5.2.1). Dès lors, il apparaît naturellement que les protéines qui possèdent une grande affinité pour les zones tubulaires sont transportées plus rapidement (voir flèche 4 sur la Fig. 5.8) que celles dont l'affinité est maximale dans la région médiane des saccules (voir flèche 5 sur la figure).

Étudions le mécanisme de création des grains de sécrétion proposé dans le modèle du flux continu de membranes. Rappelons que dans le modèle vésiculaire, de même que dans le modèle de la maturation, le bourgeonnement des grains de sécrétion est initié par des activateurs enchassés dans la membrane du saccule *trans*. Ici, il n'est plus question de bourgeonnement. Dans le modèle du flux continu de membranes, on suppose que la face *trans* de l'appareil de Golgi est purement tubulaire. Cette nouvelle donnée est motivée par les micrographes qui, comme celui proposé en Fig. 5.10, font apparaître des saccules perforés (voir P sur la figure). Ainsi, selon les biologistes, la tubularisation de la région *trans* est la conséquence d'une perforation progressive des saccules depuis la région *cis* vers la région *trans* (voir Fig. 5.8). La formation des grains de sécrétion peut alors être vue comme le résultat du désagrègement, par pincement, du saccule tubulaire.

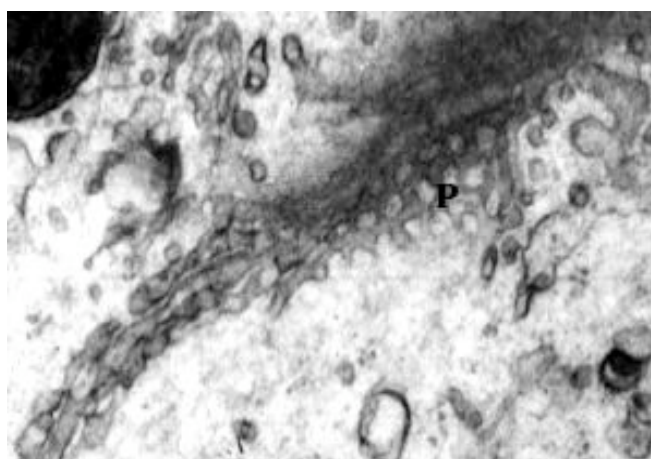


FIGURE 5.10 – Micrographe où l'on observe une vue oblique d'un appareil de Golgi d'une cellule Sertoli de rat [Rambourg 79], avec l'aimable autorisation d'Alain Rambourg (Institut IBITEC-S, CEA-Saclay)

Illustrons la création des grains de sécrétion à l'aide de la Fig. 5.11. Le maillage de la Fig. 5.11(a) représente le saccule *trans* tubulaire, résultat d'une perforation progressive. En appliquant des pincements locaux sur le réseau de tubes (graphiquement, les pincements sont représentés par des traits discontinus), on remarque l'apparition de gonflements. Les gonflements coïncident avec une concentration accrue des molécules aux abords des portions de tube pincées, ils sont à l'origine de la forme arrondie que prennent généralement les grains de sécrétion. Lorsque la pression exercée par les pincements devient trop impor-

tante, les tubes sont sectionnés. Les mécanismes de pincement de compartiments sont bien connus en biologie, ils sont le fait de protéines membranaires qui courbent la membrane [Képès 04].

Sur la Fig. 5.11(b), suite au sectionnement du réseau de tubes, trois grains de sécrétion apparaissent. On remarque que leur forme diffère en fonction de la position des pincements. Cette variété des formes que peuvent prendre les grains de sécrétion confirme les observations faites par les biologistes sur certains micrographes. Notons que le modèle du flux continu de membranes est le seul, parmi les trois modèles décrits dans ce chapitre, à proposer une piste qui permet d'expliquer la variété des formes des grains de sécrétion.

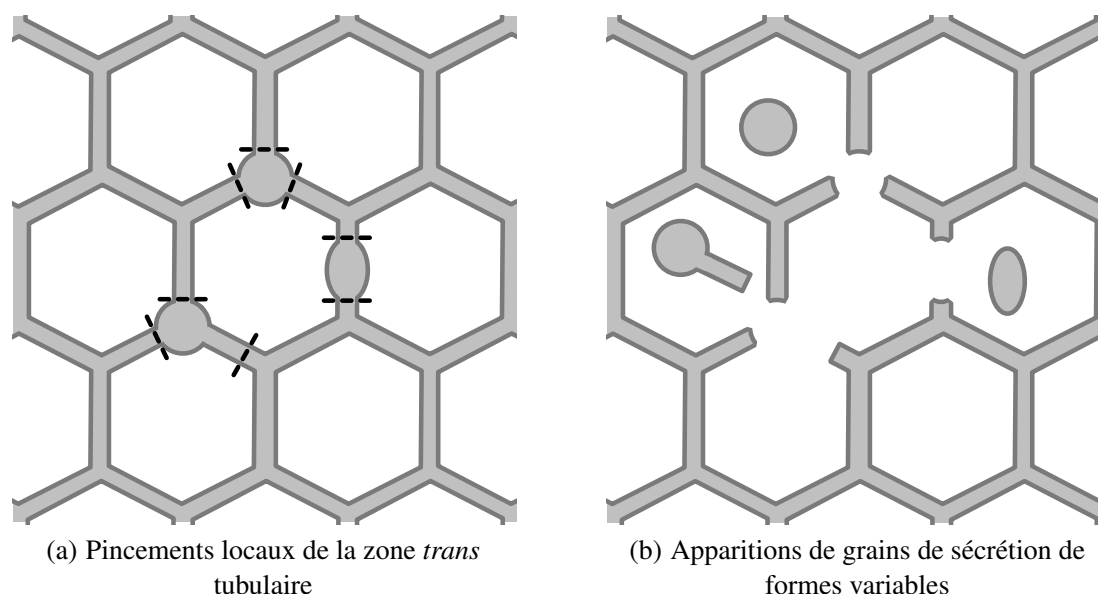


FIGURE 5.11 – Création des grains de sécrétion

Concluons sur le modèle du flux continu de membranes en ajoutant que comme dans le cas du modèle de la maturation, les enzymes responsables de la maturation des protéines migrent de la région *trans* vers la région *cis*. De cette manière, elles restent en moyenne stationnaires dans la région *cis*. Dans ce nouveau modèle cependant, leur retour n'est pas le fait de transport vésiculaire, mais le fait de transport rétrograde le long des tubes qui connectent les saccules.

Enfin, dans l'hypothèse d'un appareil de Golgi entièrement connexe, on suppose que le RE est également connecté à la région *cis*. Ainsi, le désagrègement de la région *trans*, conséquence de la formation des grains de sécrétion, est compensé par l'arrivée de matière *via* un flux continu depuis le RE vers *cis*. C'est par ce même chemin que les protéines entrent dans l'appareil de Golgi.

5.3 Bilan

Dans ce chapitre, nous avons vu que l'observation seule de l'appareil de Golgi – par microscopie optique ou électronique – ne permet pas de conclure sur sa structure. On remarque notamment que les différents modèles de l'appareil de Golgi présentent des différences nettes du point de vue topologique. Dans le modèle vésiculaire et le modèle de la maturation, les saccules sont déconnectés et des vésicules expliquent le transport des protéines. À l'inverse, dans le modèle du flux continu de membranes, les saccules sont connectés par des tubes qui permettent le transport des protéines. Les trois modèles se distinguent sensiblement par la nature de leurs compartiments ainsi que par les relations de voisinages qui

existent entre leurs compartiments.

Ainsi, en réponse à une demande des biologistes, nous souhaitons proposer un outil d'aide à l'analyse de ces modèles où l'on se focalise sur les aspects topologiques. Dans les chapitres suivants, nous proposons une méthodologie de discrimination pour les modèles biologiques où la topologie joue un rôle essentiel. Nous posons également les bases de notre outil en nous appuyant sur le développement d'un premier prototype.

MÉTHODE DE DISCRIMINATION POUR L'APPAREIL DE GOLGI

Sommaire

6.1	Formulaire	134
6.2	Notre méthode de discrimination de modèles	135
6.2.1	Discriminer par simulation	135
6.2.2	Méthodologie de discrimination de modèles	137
6.3	Modèle vésiculaire	138
6.3.1	Le modèle initial	138
6.3.2	Mise à jour du modèle vésiculaire	144
6.3.3	De la nécessité d'une zone tubulaire	148
6.3.4	Bilan	153
6.4	Modèle de la maturation	154
6.4.1	Un transport rétrograde coûteux	154
6.4.2	Bilan	156
6.5	Modèle du flux continu de membranes	156
6.5.1	De l'insuffisance de l'étude des rapports de volume sur surface	156
6.5.2	Représentations tridimensionnelles des modèles de l'appareil de Golgi	159
6.5.3	Validation des modèles tridimensionnels par des coupes	160
6.5.4	Bilan	162
6.6	Conclusion de notre étude	162

Les travaux présentés dans ce chapitre sont le résultat d'une collaboration avec des biologistes dans le cadre du groupe de travail GolgiTop¹. Ce groupe, créé en 2005, se réunit dans le cadre des ateliers de réflexion proposés par le Programme d'Épigénomique (Génopole, Évry, 91). Ce programme, créé en 2002, a pour principal objectif de fournir un support – notamment au travers d'événements scientifiques – pour le développement de collaborations entre chercheurs issus de plusieurs disciplines (principalement des biologistes, informaticiens, mathématiciens et physiciens) qui travaillent autour d'une thématique biologique commune. Dans notre cas, nous nous sommes intéressés à l'appareil de Golgi.

Le but de notre collaboration au sein de GolgiTop est d'utiliser la modélisation topologique pour l'étude d'objets biologiques structurés. Nos premières discussions nous ont rapidement orientés vers l'appareil de Golgi qui, comme nous l'avons vu dans le chapitre 5, est un objet biologique dont la structure – c'est-à-dire la topologie – et la dynamique sont encore mal connues. Dans ce même chapitre, nous avons remarqué que ces connaissances imprécises de l'appareil de Golgi sont à l'origine de plusieurs modèles parmi lesquels le modèle vésiculaire, le modèle de la maturation et le modèle du flux continu de membranes. Notre étude de l'appareil de Golgi, au sein de GolgiTop, porte particulièrement sur l'existence de tubes qui connecteraient les saccules.

Ainsi, conformément aux besoins définis par nos collègues biologistes, nous souhaitons proposer une méthode qui permet de démontrer l'importance des tubes dans l'appareil de Golgi. Notamment, nous montrerons que le modèle vésiculaire ainsi que les modèles de la maturation ne sont pas parfaitement conformes aux observations biologiques en l'état. Il est nécessaire d'adapter ces deux modèles afin de mieux coïncider avec les observations. Nous verrons que les mises à jour qui nous semblent les plus plausibles incluent la présence de tubes.

À terme, l'objectif de nos travaux au sein de GolgiTop est de proposer des outils basés sur la topologie pour l'aide à la discrimination de modèles biologiques. Même si dans le cadre de ce manuscrit, nous nous limitons à l'étude des tubes dans l'appareil de Golgi, nous veillons à définir le cahier des charges général de ce que serait un outil général de discrimination de modèles où la structure du modèle apparaît décisive.

En section 6.1, nous rappelons quelques formules de calcul de volume et de surface utiles pour la lecture de ce chapitre. Notre méthode générale de discrimination est présentée en section 6.2. Dans les sections 6.3 et 6.4 nous mettons en œuvre notre méthode dans les cas du modèle vésiculaire et de celui de la maturation. Dans ces deux cas, nous utilisons des critères de discrimination simples basés sur les dimensions – au sens des mesures – des objets étudiés. Enfin, en section 6.5, nous soulignons l'importance d'une modélisation fine de la topologie dans le cas du modèle du flux continu de membranes.

Notons que les travaux que nous exposons dans ce chapitre ont fait l'objet de trois publications [Poudret 08b, Poudret 07a, Poudret 08c].

6.1 Formulaire

Dans les calculs que nous effectuons tout au long de ce chapitre, nous apparentons les vésicules et grains de sécrétion à des sphères et les tubes à des cylindres. Afin de faciliter la lecture de ce chapitre,

1. Nous tenons particulièrement à remercier :

- Jean-Marc Verbavatz, biologiste au Laboratoire Trafic Membranaire (LTM) de l'Institut de biologie et de technologies de Saclay (iBiTec-S, CEA, 91) est responsable de l'équipe Imagerie cellulaire et moléculaire dédiée à l'étude de la voie sécrétoire ;
- Alain Rambourg, également biologiste au LTM, est conseiller scientifique dans l'équipe Imagerie cellulaire et moléculaire ;
- François Képès, directeur de recherche en Biologie au CNRS, dirige le Programme d'Épigénomique.

nous rappelons les formules de calcul de volume et de surface d'une sphère et d'un cylindre.

Le volume – intérieur – V_{sph} et la surface – extérieure – S_{sph} d'une sphère de diamètre D_{sph} (voir Fig. 6.1(a)) sont donnés par les relations suivantes :

$$V_{sph} = (4/3)\pi(D_{sph}/2)^3$$

$$S_{sph} = 4\pi(D_{sph}/2)^2$$

Le volume V_{cyl} et la surface S_{cyl} d'un cylindre de diamètre D_{cyl} et de longueur L_{cyl} (voir Fig. 6.1(b)) sont donnés par :

$$V_{cyl} = \pi \times (D_{cyl}/2)^2 \times L_{cyl}$$

$$S_{cyl} = 2 \times \pi \times (D_{cyl}/2) \times L_{cyl} + 2 \times \pi \times (D_{cyl}/2)^2$$

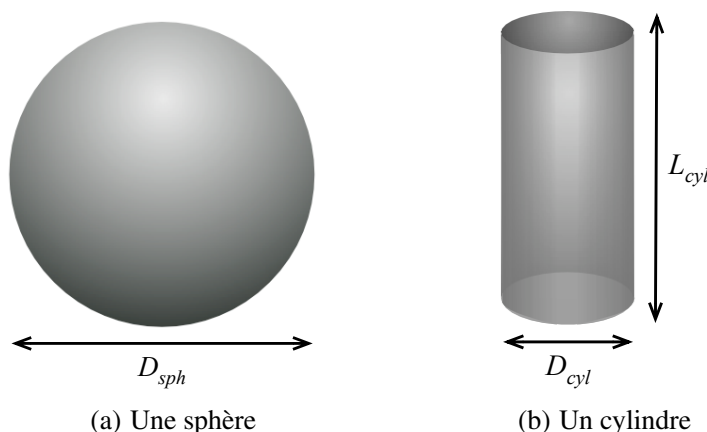


FIGURE 6.1 – Une sphère et un cylindre

6.2 Notre méthode de discrimination de modèles

6.2.1 Discriminer par simulation

Notre méthode de discrimination de modèles s'appuie sur la *simulation* des différents modèles, puis sur la comparaison des résultats de la simulation – on parle souvent de *phénomènes émergents* – avec les observations faites expérimentalement. La simulation d'un modèle biologique tel que l'appareil de Golgi est découpée en quatre grandes étapes. Nous pouvons les décrire schématiquement de la façon suivante.

Définition d'une structure statique

Intuitivement, la simulation est la mise en œuvre informatique d'un modèle théorique qui explique un phénomène réel. En amont de la simulation d'un modèle biologique, il est nécessaire de définir une structure statique qui représente la topologie du modèle étudié. Il s'agit principalement de représenter les différents compartiments biologiques qui entrent en jeu dans le modèle. Dans notre cas, nous manipulons quatre types de compartiments : les saccules, les vésicules, les grains de sécrétion et les tubes.

De plus, nous associons des données de plongement – biologiques et/ou géométriques – à chacun de ces compartiments. Par exemple, nous pouvons associer un volume et une surface à l'ensemble des

compartiments. Le choix de ces plongements est une étape décisive puisqu'il conditionne l'émergence de phénomènes que nous pouvons observer à la fin d'une simulation. Par exemple, dans l'hypothèse d'une discrimination basée sur des quantités de protéines membranaires, il est important de prévoir – parmi les données de plongement – des quantités de protéines associées à chaque membrane dans la structure.

Remarquons qu'en fonction de la complexité de l'objet étudié, il n'est pas toujours nécessaire d'utiliser un modèle géométrique à base topologique tel que les cartes généralisées (voir partie I, chapitre 1) pour représenter cette structure. Nous parlons de représentation topologique au sens général. Dans les cas simples, le choix de l'outil importe peu et une structure de données *ad hoc* peut être suffisante.

Définition d'un ensemble de règles de comportement

Afin d'animer la structure statique, et donc de représenter les aspects dynamiques du modèle étudié, nous devons définir un ensemble de *règles* – ou *lois* – de comportement. Intuitivement, une règle de comportement code un mécanisme élémentaire du modèle. Ainsi, dans le cas des modèles de l'appareil de Golgi, par exemple, des règles modélisent le bourgeonnement des vésicules à la surface des saccules tandis que d'autres modélisent la fusion des vésicules avec les saccules.

Notons que les règles de comportement peuvent modifier la structure et/ou le plongement. Par exemple, une règle qui crée un nouveau saccule dans le modèle de la maturation modifie la structure : elle ajoute un compartiment. Elle modifie également le plongement en associant un volume et une surface au nouveau saccule.

Comme lors de l'étape de définition de la structure statique, nous parlons de règles de comportement au sens général. Ainsi, lors de la simulation de modèles simples, elles ne sont pas nécessairement des règles de transformation de graphes telles que celles présentées dans la partie I.

Définition d'un ensemble de paramètres d'entrée

Au lancement d'une simulation, les données de plongement doivent toutes être initialisées. Les valeurs prises par les données de plongement au début d'une simulation sont appelées les *paramètres d'entrée*. Nous en distinguons deux types.

Les *paramètres observables* correspondent aux connaissances des biologistes. Ils sont issus de l'observation et restent constants au cours de la simulation. Dans le cas de l'appareil de Golgi, nous pouvons citer les dimensions respectives des vésicules et grains de sécrétion, mesurées expérimentalement.

Les *paramètres d'initialisation* sont choisis par l'utilisateur au début de la simulation. Ils correspondent le plus souvent à des données pour lesquelles il n'existe pas de valeur connue. C'est le cas de la dimension des saccules, qui peut varier sensiblement d'une espèce à une autre, ou même d'une image à une autre au sein d'une même espèce. Souvent, les valeurs des paramètres d'initialisation évoluent au cours de la simulation. Par exemple, en invaginant – c'est-à-dire en ingérant – des vésicules, la dimension d'un saccule peut augmenter sensiblement. Ainsi, en fin de simulation, ces paramètres sont comparés aux observations faites expérimentalement. Par exemple, dans le cas de l'appareil de Golgi, un modèle peut être réfuté s'il entraîne une croissance irréaliste des saccules.

Mise en œuvre d'une stratégie d'application des règles de comportement

À la lumière des définitions d'une structure statique, d'un ensemble de règles de comportement et d'un ensemble de paramètres d'entrée, nous pouvons définir la simulation d'un modèle biologique de la manière suivante. Il s'agit d'une boucle – appelée *boucle de simulation* – dans laquelle à chaque itération, les règles de comportements modifient la structure ainsi que les données associées aux différents

compartiments. Dans ce contexte, une *stratégie d'application* des règles de comportement définit dans quel ordre les règles de comportement sont appliquées. Dans ce chapitre, nous utilisons des stratégies simples où les règles sont appliquées en séquence.

6.2.2 Méthodologie de discrimination de modèles

Notre méthode de discrimination s'appuie sur la simulation successive de modèles biologiques. Elle est illustrée en Fig. 6.2. Pour plus de clarté, la discrimination est mise en œuvre sur deux modèles seulement, notés M_1 et M_2 , sur la figure. Intuitivement, il s'agit de déterminer lequel des deux modèles correspond le mieux aux connaissances biologiques. La figure ainsi que les commentaires associés s'étendent de manière immédiate à trois modèles, ou plus.

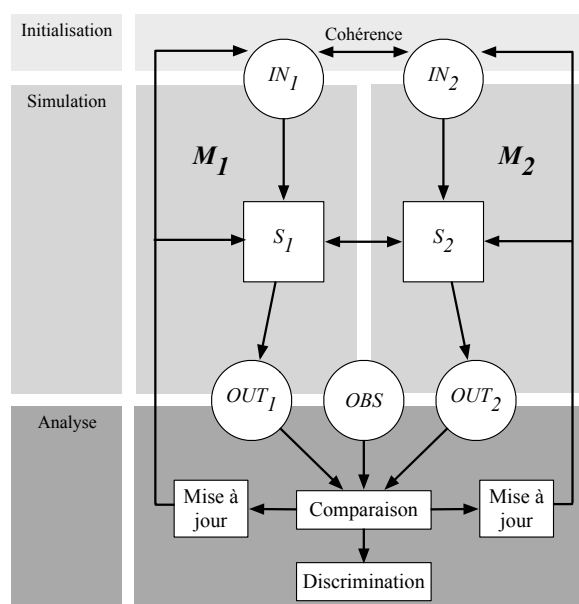


FIGURE 6.2 – Boucle de discrimination de modèles

Les paramètres d'entrée associés aux modèles M_1 et M_2 sont notés, respectivement, IN_1 et IN_2 . Ils figurent dans la partie initialisation de notre méthode. Notons que certains paramètres d'entrée sont communs aux deux modèles. Par exemple, si M_1 est le modèle vésiculaire et M_2 est le modèle du flux continu de membranes, il est pertinent de parler de dimension des saccules dans les deux modèles. Ainsi, nous devons veiller à initialiser la dimension des saccules aux mêmes valeurs dans M_1 et M_2 . Remarquons que certaines quantités n'ont de sens que dans un modèle. Par exemple, le nombre de vésicules n'a de sens que dans M_1 . Cependant, cette donnée propre à M_1 est à corrélérer au nombre de tubes qui connectent les saccules et qui n'a de sens que dans M_2 . Notons ainsi que la cohérence des paramètres de M_1 avec les paramètres de M_2 , lorsqu'elle est possible, doit toujours être respectée. Le bien-fondé de notre démarche est fortement tributaire de cette rigueur dans l'initialisation des paramètres. De la même manière, les structures statiques qui représentent la topologie des modèles M_1 et M_2 , notées respectivement S_1 et S_2 , doivent également être construites au regard de cette cohérence. Par exemple, le nombre de saccules doit être identique dans M_1 et M_2 .

Les données qui émergent de la simulation des modèles M_1 et M_2 sont notées, respectivement, OUT_1 et OUT_2 . Par exemple, la quantité de grains de sécrétion créés au cours des simulations est une donnée de sortie, c'est-à-dire un paramètre émergent – pour les deux modèles. Les dimensions des différents saccules en fin de simulation sont également une donnée de sortie. Dans la partie analyse de notre boucle de discrimination, les données de sorties sont comparées avec les données issues de l'observation expé-

rimentale, notées *OBS* sur la figure. A l'issue de cette comparaison, nous entrons dans un processus de mise à jour des deux modèles puis simulons à nouveau. L'objectif de cette mise à jour étant d'approcher les sorties OUT_1 et OUT_2 des observables *OBS*. Notons que lors de cette mise à jour, nous pouvons être amenés à re-considérer tantôt les paramètres d'entrée IN_1 et IN_2 , tantôt les structures statiques S_1 et S_2 des modèles. Dans les deux cas la cohérence des paramètres d'entrée ainsi que des structures statiques doit être préservée. La boucle de simulation et de mise à jour est répétée jusqu'à ce que l'un des deux modèles soit trop éloigné des observables et qu'il devienne impossible de faire une mise à jour qui rapprocherait sa sortie des observations. Celui-ci est alors réfuté. Notons que notre méthode n'assure pas que l'on puisse toujours conclure entre les deux modèles. C'est le cas lorsque, pour les biologistes, les deux modèles convergent de manière satisfaisante vers les observables. C'est également le cas lorsqu'aucun modèle ne permet d'approcher les connaissances biologiques. Dans les deux cas, il faut alors re-concevoir les différents modèles et repenser les différentes données, ou critères, sur lesquelles porte la discrimination.

À présent que sont posées les bases de notre méthode de discrimination de modèles biologiques, nous consacrons la suite de ce chapitre à une étude des données sur lesquelles porte la discrimination. Notamment, nous mettons en lumière le fait que des critères basés sur les rapports de volume sur surface des compartiments suffisent à démontrer l'intérêt des tubes, pourtant absents du modèle vésiculaire et du modèle de la maturation.

6.3 Modèle vésiculaire

Rappelons que dans le modèle vésiculaire (voir section 5.2.1), l'appareil de Golgi est constitué d'une pile de saccules déconnectés. La pile est statique, c'est-à-dire que les saccules sont immobiles par hypothèse. Remarquons cependant, que les saccules peuvent être sujets à des phénomènes de croissance et de décroissance. Ainsi, des compartiments moteurs, les vésicules, sont nécessaires afin de transporter les protéines depuis le réticulum endoplasmique vers les grains de sécrétion. Intuitivement, ce transport est le fait de mécanismes d'empaquetage et de dépaquetage des protéines dans de petites vésicules qui bordent la pile de saccules. Enfin, dans le modèle vésiculaire, l'excrétion des grains de sécrétion depuis la région *trans* est également modélisée par un bourgeonnement.

6.3.1 Le modèle initial

La structure statique

Traditionnellement, on distingue deux régions particulières dans l'appareil de Golgi : la région *trans* (ou face *trans*) – depuis laquelle sont excrétés les grains de sécrétion – et la région *cis* (ou face *cis*), orientée vers le réticulum endoplasmique. Dans ce document, nous choisissons de ne pas considérer les mécanismes biologiques mis en œuvre dans les transports de protéines solubles ou membranaires depuis le réticulum endoplasmique vers la région *cis*. Ainsi, nous nous focalisons sur les régions médianes et *trans* de l'appareil de Golgi.

La structure simulée est simple, elle est constituée de trois types de compartiment : les saccules, les vésicules et les grains de sécrétion. Nous distinguons trois saccules : le *Saccule 1*, le *Saccule 2* et le saccule *Trans* (situé dans la région *trans*). Ces trois saccules sont annotés sur la représentation schématique de la Fig. 6.3. Remarquons que contrairement aux saccules, le nombre de vésicules et de grains de sécrétion n'est pas fixe au cours de la simulation.

Dans cette modélisation, seules deux données sont associées à chaque compartiment : une surface et un volume. En particulier, nous ne représentons pas les protéines solubles et membranaires transportées le long de l'appareil de Golgi.

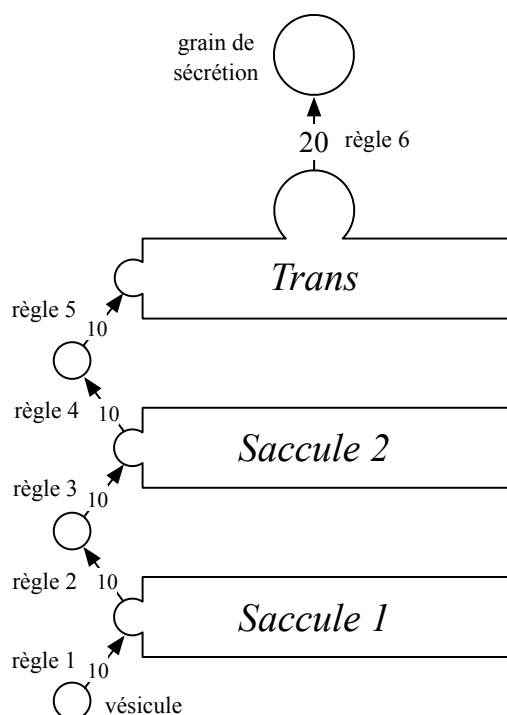


FIGURE 6.3 – Le modèle vésiculaire

La dynamique

Posons à présent l'ensemble des règles de comportement qui déterminent la dynamique du modèle vésiculaire. On distingue six comportements. Chacune de ces règles est la traduction d'un mécanisme élémentaire introduit dans le chapitre précédent :

- règle 1 : fusion de toutes les vésicules de la région *cis* avec le *Saccule 1* ;
- règle 2 : bourgeonnement de vésicules à la surface du *Saccule 1* ;
- règle 3 : fusion de toutes les vésicules issues du *Saccule 1* avec le *Saccule 2* ;
- règle 4 : bourgeonnement de vésicules à la surface du *Saccule 2* ;
- règle 5 : fusion de toutes les vésicules issues du *Saccule 2* avec le *saccule Trans* ;
- règle 6 : bourgeonnement de grains de sécrétion à la surface du *saccule Trans*.

Sur la Fig. 6.3, les règles de comportement sont représentées par des flèches. Nous choisissons la stratégie d'application des règles suivantes : les règles sont appelées en séquence dans l'ordre induit par leur numérotation.

Notons que les règles 1 à 5 manipulent simultanément plusieurs vésicules. Par exemple, la règle 1 (respectivement les règles 3 et 5) fusionne toutes les vésicules situées entre la région *cis* – qui rappelons-le n'est pas représentée dans la structure statique – et le *Saccule 1* (respectivement les vésicules situées entre les saccules 1 et 2 et celles situées entre les saccules 2 et *trans*). De la même manière, les règles 2, 4 et 6 manipulent des ensembles de vésicules ou grains de sécrétion (dans le cas de la règle 6).

Les paramètres

Définissons à présent l'ensemble des paramètres d'entrée de notre représentation du modèle vésiculaire. Puisque nous manipulons uniquement deux types de données numériques, des surfaces et des volumes, les paramètres d'entrée sont les surfaces et volumes initiaux des différents compartiments :

sacculés, vésicules et grains de sécrétion.

Déterminons dans un premier temps l'ensemble des paramètres observables, rappelons qu'ils correspondent aux connaissances biologiques et restent constants au cours de la simulation. Dans notre cas, ce sont les dimensions des vésicules et des grains de sécrétion. Pour ceux-ci, de nombreuses observations expérimentales existent (voir, par exemple, Fig. 5.3, page 124). Notamment, ils semblent généralement de forme sphérique et leur diamètre est bien connu. Ainsi, sur les images issues de la microscopie électronique, le diamètre d'une vésicule est d'environ $D_{ves} \simeq 60 \text{ nm}$. Les calculs du volume, de la surface et du rapport de volume sur surface d'une vésicule – notés respectivement V_{ves} , S_{ves} et R_{ves} – sont immédiats lorsque l'on considère que les vésicules sont des sphères :

$$\begin{aligned} V_{ves} &\simeq (4/3)\pi(D_{ves}/2)^3 = 110 \times 10^3 \text{ nm}^3 \\ S_{ves} &\simeq 4\pi(D_{ves}/2)^2 = 11 \times 10^3 \text{ nm}^2 \\ R_{ves} &\simeq V_{ves}/S_{ves} = 10 \text{ nm} \end{aligned}$$

Le diamètre d'un grain de sécrétion est d'environ $D_{gra} \simeq 2 \times 60 \simeq 120 \text{ nm}$. Son volume V_{gra} , sa surface S_{gra} et son rapport de volume sur surface R_{gra} sont alors :

$$\begin{aligned} V_{gra} &\simeq (4/3)\pi(D_{gra}/2)^3 = 900 \times 10^3 \text{ nm}^3 \\ S_{gra} &\simeq 4\pi(D_{gra}/2)^2 = 45 \times 10^3 \text{ nm}^2 \\ R_{gra} &\simeq V_{gra}/S_{gra} = 20 \text{ nm} \end{aligned}$$

Au regard de R_{ves} et R_{gra} , on remarque que les vésicules et les grains de sécrétion ne présentent pas le même rapport de volume sur surface. Nous considérons que la surface d'un compartiment est corrélée à sa quantité de membrane. Intuitivement, plus un compartiment présente une surface de dimension importante, plus il possède de membrane. Ainsi, les différences de rapport de volume sur surface des vésicules et des grains de sécrétion nous indiquent que les grains de sécrétion présentent un déficit en surface, c'est-à-dire en membranes, comparés aux vésicules. Sur la Fig. 6.3, les rapports de volume sur surface étiquettent les flèches. Par exemple, la flèche étiquetée par 20 indique que les grains de sécrétion excrétés depuis le sacculé *Trans* sont de rapport volume sur surface 20 nm. L'ensemble des grandeurs relatives aux vésicules et grains de sécrétion est synthétisé dans le Tab. 6.1.

Paramètre	Valeur
Diamètre d'une vésicule	$D_{ves} \simeq 60 \text{ nm}$
Volume d'une vésicule	$V_{ves} \simeq 110 \times 10^3 \text{ nm}^3$
Surface d'une vésicule	$S_{ves} \simeq 11 \times 10^3 \text{ nm}^2$
Rapport volume/surface d'une vésicule	$R_{ves} \simeq 10 \text{ nm}$
Diamètre d'un grain de sécrétion	$D_{gra} \simeq 120 \text{ nm}$
Volume d'un grain de sécrétion	$V_{gra} \simeq 900 \times 10^3 \text{ nm}^3$
Surface d'un grain de sécrétion	$S_{gra} \simeq 45 \times 10^3 \text{ nm}^2$
Rapport volume/surface d'un grain de sécrétion	$R_{gra} \simeq 20 \text{ nm}$

Tab 6.1 – Paramètres observables

Déterminons à présent les données qui doivent être initialisées par l'utilisateur. Dans le cas du modèle vésiculaire, celles-ci sont :

- le nombre de vésicules transportées depuis la région *cis* vers le *Saccule 1* à chaque pas de simulation. Remarquons que ce paramètre est lié à la règle de comportement 1. Il est corrélé au flux de matière qui entre dans l'appareil de Golgi par la région *cis* ;
- le volume initial des saccules ;
- la surface initiale des saccules.

Un dernier paramètre d'initialisation est lié aux règles de bourgeonnement des vésicules et des grains de sécrétion (règles 2, 4 et 6). Les biologistes estiment que les mécanismes de bourgeonnement sont déclenchés par des molécules, que nous appelons *activateurs*, enchâssées dans la membrane des saccules. Nous considérons que chaque activateur permet le bourgeonnement d'exactly une vésicule – ou grain de sécrétion – à chaque itération. Comme les activateurs sont enchâssés dans la membrane des saccules, plus un saccule contient de membrane – c'est-à-dire plus sa surface est grande – plus il présente d'activateurs. Par conséquent, plus la surface d'un saccule est grande, plus la quantité de vésicules et de grains de sécrétion excrétée à chaque itération est élevée. Nous proposons d'ajouter un paramètre d'initialisation complémentaire qui représente la quantité de surface associée aux activateurs, modélisant ainsi la densité des activateurs le long de la membrane. Nous supposons que cette quantité est identique pour tous les saccules, qu'ils excrètent des vésicules ou des grains de sécrétion.

Notons enfin que nous ne proposons pas de facteur limitant afin de borner le nombre de vésicules qui fusionnent avec un saccule au cours d'une itération (voir les règles de comportement 2 et 3). Intuitivement, à chaque itération, toutes les vésicules situées entre deux saccules fusionnent avec le saccule supérieur.

Trois simulations

Dans cette section, nous proposons trois simulations du modèle vésiculaire, correspondant à trois jeux de paramètres d'initialisation, introduits dans le Tab. 6.2. On remarque que les quantités de volume et de surface initiales des saccules sont nulles. Cependant, le nombre de vésicules transportées vers le *Saccule 1* à chaque itération, de même que la quantité de surface associée aux activateurs, varie d'une simulation à une autre.

Simulation	Paramètre	Valeur
Simulation 1	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Saccule 1</i> à chaque itération	10
	Surface par activateur	$36 \times 10^4 \text{ nm}^2$
Simulation 2	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Saccule 1</i> à chaque itération	5
	Surface par activateur	$36 \times 10^4 \text{ nm}^2$
Simulation 3	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Saccule 1</i> à chaque itération	10
	Surface par activateur	$18 \times 10^4 \text{ nm}^2$

Tab 6.2 – Paramétrage de trois simulations

Étudions à présent les résultats de chaque simulation illustrés par les graphiques de la Fig. 6.4. Les Fig. 6.4(a) et 6.4(b) illustrent respectivement l'évolution de la surface et du volume des saccules au cours de la simulation 1. Les courbes épaisse noire, épaisse grise et fine noire correspondent respectivement au *Saccule 1*, *Saccule 2* et saccule *Trans* (cette convention est conservée dans tout le chapitre). De la même manière, les Fig. 6.4(c) et 6.4(d) correspondent à la simulation 2 tandis que les Fig. 6.4(e) et 6.4(f) correspondent à la simulation 3.

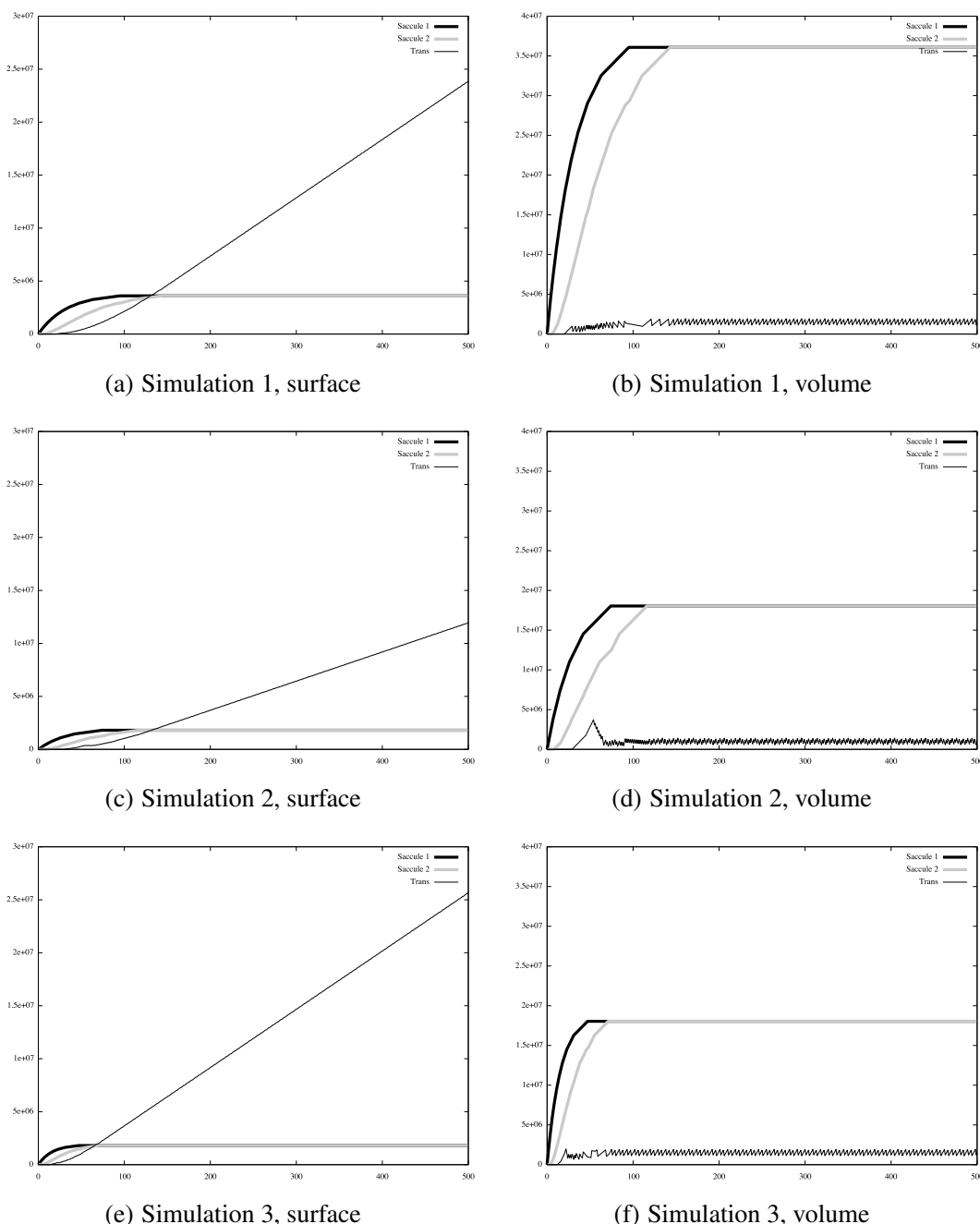


FIGURE 6.4 – Résultat des simulations 1, 2 et 3

Sur la Fig. 6.4(a), on remarque que la quantité de surface du *Saccule 1* converge vers $360 \times 10^4 \text{ nm}^2$ après 100 itérations. Intuitivement, la quantité de surface d'un saccule converge lorsque la « surface excrétée » – c'est-à-dire la quantité de membranes qui sort du saccule – est égale à la « surface fusion-

néen », c'est-à-dire la quantité de membranes qui entre dans le saccule. Dans le cas du *Saccule 1*, la surface excrétée et la surface fusionnée sont égales lorsqu'il y a équilibre entre le nombre de vésicules sortantes et entrantes. Le Tab. 6.2 nous indique qu'à chaque itération, 10 vésicules fusionnent avec le *Saccule 1*. Sachant qu'une surface de $36 \times 10^4 \text{ nm}^2$ est associée à chaque activateur, pour excréter 10 vésicules, la surface du *Saccule 1* doit être de $10 \times (36 \times 10^4) \text{ nm}^2$, soit $360 \times 10^4 \text{ nm}^2$: la valeur de convergence.

On remarque que pour les mêmes raisons, la surface du *Saccule 2* converge également vers $360 \times 10^4 \text{ nm}^2$. La surface est stable au bout de 150 itérations, contre 100 pour le *Saccule 1*. Ce décalage est dû au délai induit par l'excrétion des vésicules depuis le *Saccule 1*. La surface des saccules 1 et 2 se comporte de manière analogue dans les simulations 2 et 3 (voir, respectivement, Fig. 6.4(c) et 6.4(e)), avec toutefois des valeurs de convergence différentes en raison des variations apportées aux paramètres d'initialisation.

Étudions à présent l'évolution de la surface du saccule *Trans*. On remarque que contrairement aux *Saccule 1* et *Saccule 2*, il n'y a pas de convergence. La surface du saccule *Trans* augmente linéairement au cours du temps. Le déséquilibre entre le rapport de volume sur surface des vésicules et des grains de sécrétion explique l'augmentation de la surface dans la région *Trans*. En effet, sur la Fig. 6.3, il est indiqué que le rapport de volume sur surface des compartiments sortant du saccule *Trans* (les grains de sécrétion) est deux fois plus important que le rapport des compartiments entrants (les vésicules). Or, nous avons vu que les grains de sécrétion comportent un déficit en surface par rapport aux vésicules. La conséquence de ce déséquilibre est une accumulation de membrane, qui se traduit par une augmentation de la surface, dans la région *trans*. Le volume d'un grain de sécrétion est d'environ 8 fois celui d'une vésicule tandis que sa surface est d'environ 4 fois celle d'une vésicule. Ainsi, chaque grain de sécrétion qui bourgeonne conduit à un stockage, dans le saccule *Trans*, de $8 - 4 = 4$ fois la surface d'une vésicule. Ici encore, on remarque que la surface du saccule *Trans* augmente de la même manière dans les simulations 2 et 3.

Sur la Fig. 6.4(b), qui décrit l'évolution du volume des saccules, on note que le volume du saccule *Trans* est faible. En effet, le nombre de grains de sécrétion excrétés est borné par le volume du saccule *Trans*, faible devant la quantité de surface. Remarquons que la quantité de surface associée aux activateurs du saccule *Trans*, c'est-à-dire aux activateurs responsables du bourgeonnement des grains de sécrétion, ne borne pas le bourgeonnement des grains de sécrétion. Si tel était le cas, on noterait également une accumulation de volume dans la région *trans*. C'est en fait le volume du saccule *Trans* qui limite le bourgeonnement des grains de sécrétion.

Sur cette même figure, on remarque que le volume des *Saccule 1* et *Saccule 2* converge après, respectivement, 100 et 150 itérations ; c'est-à-dire lorsqu'il y a équilibre entre les vésicules entrantes et sortantes. Les volumes des saccules évoluent de la même manière dans les simulations 2 et 3 (voir, respectivement, Fig. 6.4(d) et Fig. 6.4(f)).

Bilan des simulations 1, 2 et 3

Les simulations 1, 2 et 3 montrent que pour trois jeux de paramètres différents, le modèle vésiculaire conduit à une croissance linéaire de la surface en *Trans*. Dans les trois simulations, cette croissance s'explique par un déséquilibre entre les rapports de volume sur surface des compartiments qui entrent et sortent du saccule *trans*. Ce phénomène est incompatible avec les observations expérimentales. Il discrédite donc le modèle vésiculaire, en l'état. Notons que les arguments avancés – centrés sur une étude des rapports de volume sur surface – indiquent que la croissance du saccule *Trans* serait également observée pour des jeux de paramètres différents de ceux proposés dans les trois simulations.

Remarquons que dans cette étude, nous nous appuyons sur une structure statique simple : seuls quelques compartiments sont représentés, et seulement deux données – une surface et un volume – sont associées à chaque compartiment. Nous avons effectué les différentes simulations en utilisant une représentation à base de cartes généralisées de dimension 3. Cependant, étant donné l'extrême simplicité de la structure simulée, l'utilisation des cartes généralisées apporte peu. Notamment, nous aurions pu calculer l'évolution de la structure à l'aide d'une technique de modélisation à base d'équations différentielles. Cependant, il ne s'agit pas tant de montrer l'importance de l'outil que de montrer que grâce à nos collaborations avec des experts biologistes du domaine, cette structure simple suffit à invalider le modèle vésiculaire tel qu'il est habituellement présenté dans la littérature (voir section 5.2.1).

6.3.2 Mise à jour du modèle vésiculaire

Mise à jour des règles de comportement et des paramètres d'entrée

Afin de rendre le modèle vésiculaire réaliste, il convient d'éliminer la croissance linéaire de la surface – c'est-à-dire l'augmentation de la quantité de membranes – dans le saccule *Trans*. Une solution immédiate consiste à dégrader – c'est-à-dire retirer – continuellement une partie de la membrane du saccule *Trans*. Pour cela, nous proposons de raffiner le modèle vésiculaire original en ajoutant une nouvelle règle de comportement, qui pilote cette dégradation. Nous conservons la structure statique introduite dans la section précédente.

Les règles de comportement considérées sont, pour la plupart, issues des simulations proposées dans la section précédente. Seule la règle 7 est nouvelle. L'ensemble des règles mises en jeu devient alors :

- règle 1 : fusion de toutes les vésicules de la région *cis* avec le *Saccule 1* ;
- règle 2 : bourgeonnement de vésicules à la surface du *Saccule 1* ;
- règle 3 : fusion de toutes les vésicules issues du *Saccule 1* avec le *Saccule 2* ;
- règle 4 : bourgeonnement de vésicules à la surface du *Saccule 2* ;
- règle 5 : fusion de toutes les vésicules issues du *Saccule 2* avec le saccule *Trans* ;
- règle 6 : bourgeonnement de grains de sécrétion à la surface du saccule *Trans* ;
- règle 7 : dégradation d'une portion de la surface du saccule *Trans*.

Un nouveau paramètre d'initialisation, associé à la règle 7, détermine quelle proportion de la surface du saccule *Trans* est dégradée à chaque itération. À nouveau, au cours d'une boucle de simulation, les règles sont appelées en séquence dans l'ordre induit par leur numérotation.

Trois simulations avec dégradation de la surface du saccule *Trans*

Nous proposons à nouveau trois simulations du modèle vésiculaire, avec cette fois-ci une dégradation de la surface du saccule *Trans*. Nous les numérotons 4, 5 et 6. Les paramètres observables sont identiques à ceux introduits lors des simulations 1, 2 et 3 (voir Tab 6.1, page 140). Les paramètres d'initialisation sont introduits dans le Tab. 6.3. Afin d'étudier l'effet de la dégradation du saccule *Trans* sur la simulation, nous choisissons de faire varier la proportion de surface dégradée. Notons que comme précédemment, dans les trois simulations, les quantités de volume et de surface initiales des saccules sont nulles. Enfin, le nombre de vésicules transportées vers le *Saccule 1* ainsi que la surface associée à chaque activateur sont identiques pour les trois simulations : respectivement 10 vésicules et $36 \times 10^4 \text{ nm}^2$ (comme dans la simulation 1, voir Tab. 6.2, page 141).

Les résultats des simulations 4, 5 et 6 sont illustrés par les graphiques de la Fig. 6.5. Les Fig. 6.5(a) et 6.5(b) illustrent respectivement l'évolution de la surface des saccules ainsi que l'évolution de la surface dégradée en *trans* au cours de la simulation 4. De la même manière, les Fig. 6.5(c) et 6.5(d) corres-

Simulation	Paramètre	Valeur
Simulation 4	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Saccule 1</i> à chaque itération	10
	Surface par activateur	$36 \times 10^4 \text{ nm}^2$
	Surface dégradée en <i>trans</i> à chaque itération	3 %
Simulation 5	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Saccule 1</i> à chaque itération	10
	Surface par activateur	$36 \times 10^4 \text{ nm}^2$
	Surface dégradée en <i>trans</i> à chaque itération	1.5 %
Simulation 6	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Saccule 1</i> à chaque itération	10
	Surface par activateur	$36 \times 10^4 \text{ nm}^2$
	Surface dégradée en <i>trans</i> à chaque itération	0.5 %

Tab 6.3 – Paramétrage de trois simulations avec dégradation de surface en *trans*

pondent à la simulation 5 tandis que les Fig. 6.5(e) et 6.5(f) correspondent à la simulation 6.

Le Tab. 6.3 nous indique que la règle de comportement 7 induit une dégradation de 3 % de la surface du saccule *Trans* à chaque itération. Comme la dégradation n'influe pas sur les *Saccule 1* et *Saccule 2*, les courbes épaisse noire et épaisse grise de la Fig. 6.5(a) sont identiques aux courbes de la première simulation (illustrées en Fig. 6.4(a), page 142). La convergence s'explique à nouveau par l'équilibre des entrées et sorties dans les saccules (voir section 6.3.1).

Remarquons qu'à l'opposé des simulations 1, 2 et 3, la surface du saccule *Trans* converge à partir d'environ 200 itérations. Illustrons les mécanismes mis en œuvre dans l'établissement de cet état stationnaire à l'aide de la Fig. 6.6. Comme le *Saccule 1* est stationnaire, il existe un équilibre entre les vésicules qui entrent et sortent du saccule (voir section 6.3.1). Le Tab. 6.3 nous indique que 10 vésicules sont transportées depuis le réticulum endoplasmique vers le *Saccule 1* à chaque itération. Ainsi, à l'équilibre, 10 vésicules sont également excrétées du *Saccule 1* à chaque itération et fusionnent avec le *Saccule 2*. Par un raisonnement analogue, on établit que 10 vésicules fusionnent avec le saccule *Trans* à chaque itération (voir la flèche orientée vers le saccule *Trans* sur la Fig. 6.6).

Considérant les dimensions respectives des vésicules et des grains de sécrétion, calculons le nombre n_{gra} de grains de sécrétion que l'on peut créer avec 10 vésicules. Rappelons que le rapport de volume sur surface $R_{ves} \simeq 10 \text{ nm}$ des vésicules est faible devant le rapport $R_{gra} \simeq 20 \text{ nm}$ des grains de sécrétions. Ainsi, intuitivement, les vésicules possèdent un excédant de membranes par rapport aux grains de sécrétion. La quantité n_{gra} est donc bornée par le volume des vésicules. Dans le calcul suivant, V_{ves} et V_{gra} sont respectivement le volume d'une vésicule et d'un grain de sécrétion :

$$n_{gra} = (10 \times V_{ves})/V_{gra} \simeq 1.2$$

On obtient que 10 vésicules permettent de créer 1.2 grains de sécrétion (voir la flèche orientée vers le grain de sécrétion sur la figure). Cependant, on note que si l'on retire l'équivalent d'1.2 grains de

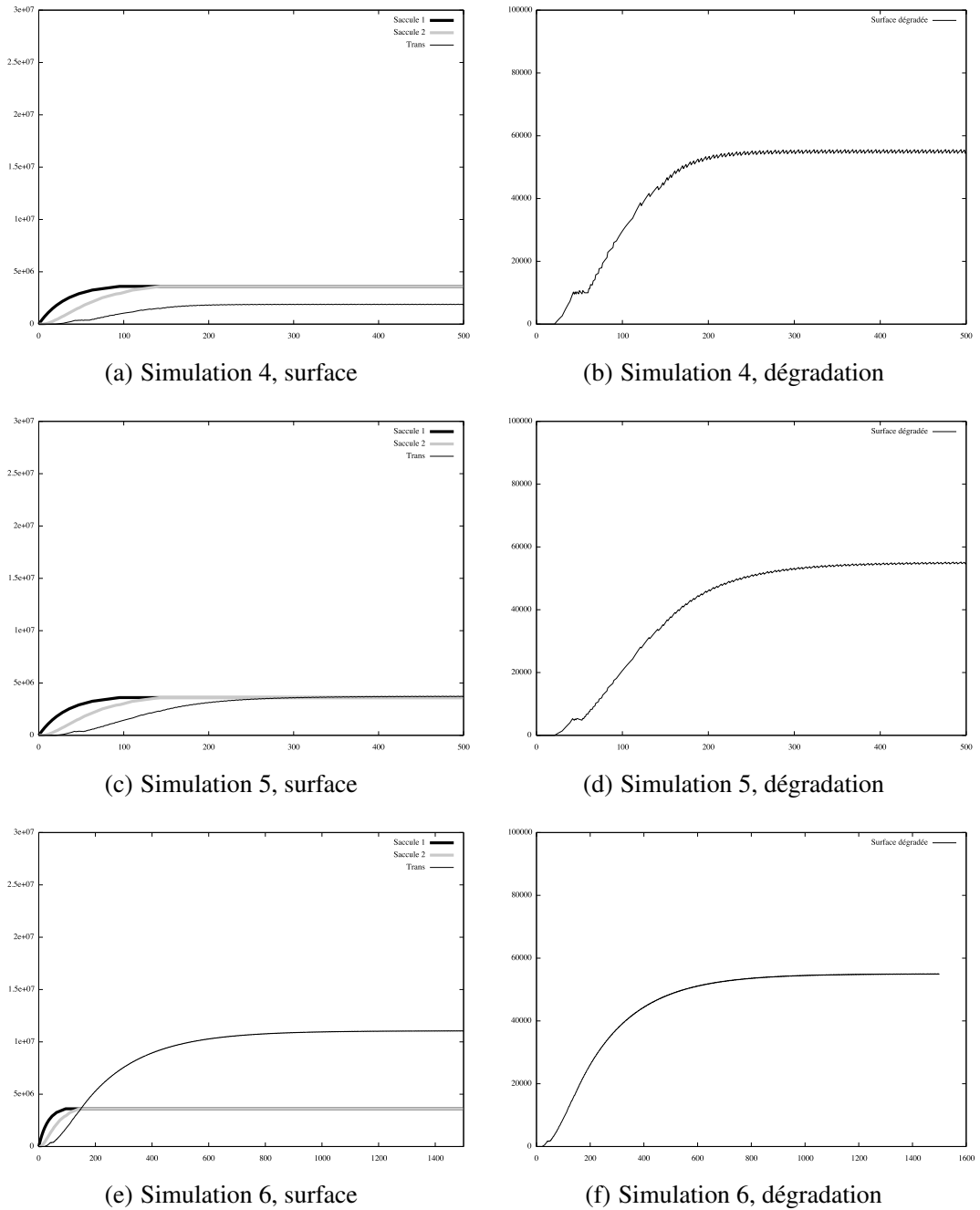


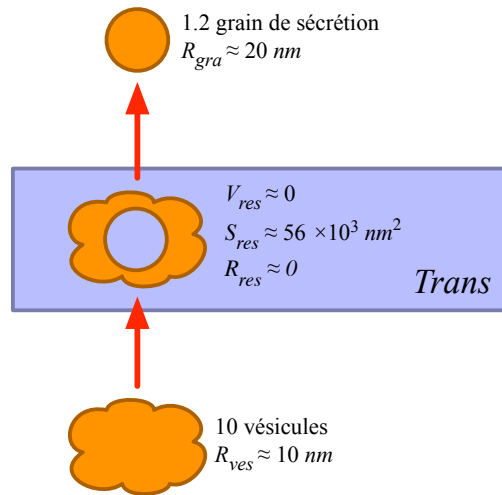
FIGURE 6.5 – Résultat des simulations 4, 5 et 6

sécrétion – en surface et volume – à l’ensemble de 10 vésicules qui entrent en *trans*, il subsiste une structure résiduelle de volume V_{res} et de surface S_{res} et de rapport R_{res} dans le saccule *Trans*. Dans le calcul suivant, S_{ves} et S_{gra} sont respectivement la surface d’une vésicule et d’un grain de sécrétion :

$$\begin{aligned}
 V_{res} &= (10 \times V_{ves}) - n_{gra} \times V_{gra} \\
 &= (10 \times V_{ves}) - ((10 \times V_{ves})/V_{gra}) \times V_{gra} \\
 &= 0
 \end{aligned}$$

$$S_{res} = (10 \times S_{ves}) - n_{gra} \times S_{gra} \simeq 56 \times 10^3 \text{ nm}^2$$

$$R_{res} = V_{res}/S_{res} = 0$$

FIGURE 6.6 – Principe de la dégradation de surface en *trans*

La structure résiduelle apparaît dans le saccule *Trans*, sur la Fig. 6.6. Comme indiqué dans la section 6.3.1, son rapport de volume sur surface $R_{res} \simeq 0$ – faible devant le rapport des vésicules $R_{ves} \simeq 10nm$ et des grains de sécrétion $R_{gra} \simeq 20nm$ – induit une accumulation de membrane dans le saccule *Trans*.

Rappelons qu’afin d’éviter une accumulation de membranes dans le saccule *Trans*, nous avons mis en jeu une règle de dégradation de la surface en *Trans*. La surface S_{res} de la structure résiduelle nous indique qu’il faut dégrader environ $56 \times 10^3 nm^2$ de surface à chaque itération afin de stabiliser la surface du saccule *Trans*. Cette valeur est confirmée par la courbe de la Fig. 6.5(b). On remarque qu’à l’itération 200, date à laquelle la quantité de surface du saccule *Trans* converge (voir Fig. 6.5(a)), la quantité de surface dégradée converge vers environ $56 \times 10^3 nm^2$ (voir Fig. 6.5(b)).

Notons enfin que puisque 3 % de la surface du saccule *Trans* est dégradée à chaque itération, le saccule *Trans* atteint l’équilibre lorsque sa quantité de surface est d’environ $187 \times 10^4 nm^2$ (puisque 3 % de $187 \times 10^4 nm^2$ font $56 \times 10^3 nm^2$).

Sur les Fig. 6.5(c) et Fig. 6.5(e), on remarque que la surface du saccule *Trans* converge également lors des simulations 5 et 6. Remarquons que même si dans nos trois nouvelles simulations, la surface en *trans* converge lorsque $56 \times 10^5 nm^2$ de surface est dégradé, les valeurs de convergence de la surface du saccule *Trans* diffèrent. Ceci est dû à la proportion différente de surface dégradée dans les trois simulations (voir Tab. 6.3). Ainsi, par exemple, dans la simulation 5, la surface en *trans* converge vers $\simeq 370 \times 10^4 nm^2$ puisque 1.5% de $\simeq 370 \times 10^4 nm^2$ font $56 \times 10^5 nm^2$.

Bilan des simulations 4, 5 et 6

Les simulations 4, 5 et 6 nous éclairent sur la nécessité de dégrader une partie de la surface du saccule *Trans* afin d’obtenir un modèle stable. Cependant, se pose la question du réalisme biologique d’une voie de dégradation de surface pure sans volume – c’est-à-dire sans matière soluble – associé. Notons également qu’en raison de l’ajout d’une voie de dégradation de membrane dans la région *trans*, nous installons une distinction non négligeable entre le modèle vésiculaire original et celui simulé ici.

6.3.3 De la nécessité d'une zone tubulaire

Dans la section 6.3.2, nous avons vu que l'intérêt d'une voie de dégradation est de compenser l'accumulation de membrane dans le saccule *Trans*. D'après les biologistes, une simple voie de dégradation de surface pure depuis la région *trans* paraît peu réaliste. En effet, une surface se referme nécessairement sur elle-même pour constituer un volume. Il semble donc probable qu'un flux de volume accompagne la dégradation. En d'autres termes, l'établissement d'un équilibre dans la région *trans* serait le fait d'une production de structures présentant des quantités non nulles de surface et de volume. De plus, il apparaît que l'état stationnaire du saccule *Trans* est atteint d'autant plus rapidement que le rapport de volume sur surface de ces nouvelles structures est faible devant le rapport des grains de sécrétion. Intuitivement, plus le rapport de cette structure est faible, plus il compense le rapport élevé des grains. Notons que dans le cas d'une voie de dégradation de surface (sans volume associé), le rapport de volume sur surface est de 0.

Supposons que ces structures, qui présentent des quantités non nulles de surface et de volume, soient des vésicules. Afin de compenser les rapports de volume sur surface élevés des grains de sécrétions, $R_{gra} \simeq 20 \text{ nm}$, nous choisissons de considérer de petites vésicules de $D_{pves} \simeq 30 \text{ nm}$ de diamètre, soit la moitié du diamètre des vésicules présentées jusqu'ici. Calculons leur volume V_{pves} , leur surface S_{pves} et leur rapport de volume sur surface R_{pves} :

$$V_{pves} = (4/3)\pi(D_{pves}/2)^3 \simeq 141 \times 10^2 \text{ nm}^3$$

$$S_{pves} = 4\pi(D_{pves}/2)^2 \simeq 28 \times 10^2 \text{ nm}^2$$

$$R_{pves} = V_{pves}/S_{pves} \simeq 5 \text{ nm}$$

Calculons le nombre n_{pves} de petites vésicules nécessaires pour chaque grain de sécrétion produit afin d'équilibrer les rapports de volume sur surface des flux entrants et sortants du saccule *Trans*. Nous savons qu'afin d'équilibrer le saccule *Trans*, le rapport global des structures excrétées par le saccule *Trans* doit être de 10. Posons alors l'égalité suivante (comme précédemment, V_{gra} et S_{gra} sont respectivement le volume et la surface d'un grain de sécrétion) :

$$\begin{aligned} \frac{V_{gra} + n_{pves} \times V_{pves}}{S_{gra} + n_{pves} \times S_{pves}} &= 10 \\ n_{pves} &= \frac{10 \times S_{gra} - V_{gra}}{V_{pves} - 10 \times S_{pves}} \\ n_{pves} &\simeq 32 \end{aligned}$$

Il apparaît nécessaire de produire environ 32 vésicules de 30 nm pour chaque grain de sécrétion. En raison de la faible épaisseur d'une coupe d'appareil de Golgi issue de la microscopie électronique, on ne peut dire que 32 vésicules doivent apparaître à chaque grain de sécrétion observé. Seules quelques vésicules, parmi ces 32 apparaîtraient effectivement sur la coupe. Avant de calculer le rapport du nombre de vésicules par rapport au nombre de grains de sécrétion, nous utilisons un modèle probabiliste simplificateur.

Nous considérons qu'à la périphérie proche d'une coupe, la densité d_{gra} des grains de sécrétion – c'est-à-dire le nombre de grains par unité de volume – est constante. Il en est de même pour la densité d_{pves} des petites vésicules. Le calcul précédent nous donne :

$$d_{pves} = n_{pves} \times d_{gra}$$

Afin d'illustrer nos calculs, appuyons-nous sur la Fig. 6.7, qui schématise une coupe utilisée en microscopie électronique. Les Fig. 6.7(a) et 6.7(b) représentent respectivement une vue en perspective et

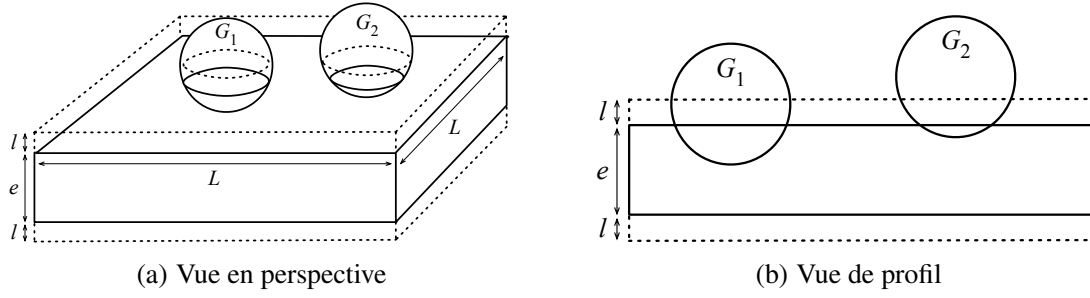


FIGURE 6.7 – Schéma d'une coupe utilisée en microscopie électronique

une vue de profil d'une telle coupe. Sur la Fig. 6.7(a), la coupe est dessinée en traits pleins, elle est de longueur L et d'épaisseur e . Les sphères enchassées dans la coupe représentent des grains de sécrétion. Afin de calculer l'espérance $E(nG)$ du nombre de grains de sécrétion observés, nous comptabilisons les grains de sécrétion dont le centre est situé dans la coupe réelle, mais également ceux dont le centre est situé juste au dessus ou en dessous de celle-ci. Pour cela, nous considérons une *coupe virtuelle* composée de la coupe réelle à laquelle on ajoute les deux volumes d'épaisseur l dessinés en pointillés. La coupe virtuelle nous permet de comptabiliser ces derniers. La Fig. 6.7(b) illustre ce principe. On remarque que le grain G_1 , dont le centre est situé dans la coupe virtuelle, mais dont le centre est en dehors de la coupe réelle, apparaît néanmoins dans la coupe réelle. Le grain G_2 , dont le centre est en dehors et de la coupe réelle et de la coupe virtuelle est trop éloigné pour être observé clairement sur la coupe réelle.

Soit V_{CoupeV} le volume de la coupe virtuelle. À l'aide de ce modèle simpliste d'une coupe, calculons à présent E_{gra} , l'espérance du nombre de grains de sécrétion dans une coupe, c'est-à-dire du nombre de grains de sécrétion réellement observés. Dans ce calcul, d_{gra} est le nombre de grains de sécrétion par unité de volume :

$$E_{gra} = d_{gra} \times V_{CoupeV} \simeq d_{gra} \times L^2 \times (e + 2 \times l)$$

En accord avec les experts biologistes, nous considérons que les vésicules de 30 nm de diamètre dont le centre est situé en dehors de la coupe virtuelle ne sont pas observées. Soit V_{Coupe} le volume de la coupe réelle. Calculons E_{pves} , l'espérance du nombre de petites vésicules observées. Dans ce calcul, d_{pves} est le nombre de petites vésicules par unité de volume :

$$E_{pves} = d_{pves} \times V_{Coupe} \simeq d_{pves} \times L^2 \times e$$

Ne reste alors qu'à calculer le rapport de l'espérance E_{pves} sur l'espérance E_{gra} . Rappelons que n_{pves} est le nombre de petites vésicules produites par grain de sécrétion afin d'équilibrer les rapports des flux entrants et sortants du saccule *Trans* :

$$E_{pves}/E_{gra} = d_{pves}/d_{gra} \times L^2/L^2 \times e/(e + 2 \times l) \simeq n_{pves} \times e/(e + 2 \times l)$$

Prenons à présent l'exemple d'une coupe d'épaisseur $e = 90 \text{ nm}$ avec $l = 30 \text{ nm}$. Nous obtenons :

$$E_{pves}/E_{gra} \simeq n_{pves} \times e/(e + 2 \times l) \simeq 19.2$$

En conclusion, pour une telle configuration de coupe, les biologistes devraient observer environ 19 fois plus de vésicules de 30 nm que de grains de sécrétion sur une coupe. Or, les biologistes n'observent pas de telles petites vésicules, qui plus est dans de telles proportions. D'une manière générale, les biologistes n'observent pas de vésicules d'un diamètre inférieur à 60 nm (ce qui n'implique pas nécessairement leur absence). L'excrétion de petites vésicules ne permet donc pas de compenser l'accumulation de membranes dans la région *trans*.

Mise à jour des règles de comportement et des paramètres d'entrée

Sur certains micrographes d'appareils de Golgi, on observe de petits tubes à la périphérie de la région *trans*. Ils présentent un diamètre moyen inférieur à 60 nm. Étudions les effets de l'excrétion de tubes dans la région *trans* de l'appareil de Golgi. Nous souhaitons examiner si les tubes permettraient, à la manière des petites vésicules, d'équilibrer les dimensions du saccule *Trans* au cours des simulations.

Nous proposons ainsi de modifier l'ensemble des règles de comportement en remplaçant la règle 7 de dégradation de surface (voir 6.3.2) par une règle de création de tubes. L'ensemble des règles mises en jeu devient alors :

- règle 1 : fusion de toutes les vésicules de la région *cis* avec le Saccule 1 ;
- règle 2 : bourgeonnement de vésicules à la surface du Saccule 1 ;
- règle 3 : fusion de toutes les vésicules issues du Saccule 1 avec le Saccule 2 ;
- règle 4 : bourgeonnement de vésicules à la surface du Saccule 2 ;
- règle 5 : fusion de toutes les vésicules issues du Saccule 2 avec le saccule *Trans* ;
- règle 6 : bourgeonnement de grains de sécrétion à la surface du saccule *Trans* ;
- règle 7 : création de tubes à la surface du saccule *Trans*.

Nous supposons que les tubes sont de forme cylindrique. Ainsi, un nouveau paramètre d'entrée, la longueur moyenne des tubes, est ajoutée et corrélée à la règle de comportement 7. On suppose que la création d'un tube depuis le saccule *Trans* est initiée par les même activateurs que ceux à l'origine des grains de sécrétion.

Deux simulations avec création de tubes à la surface du saccule *Trans*

Les résultats des simulations 7 et 8 sont illustrés par les graphiques de la Fig. 6.8. Les Fig. 6.8(a) et 6.8(b) illustrent respectivement l'évolution de la surface des saccules et l'évolution du rapport du nombre cumulé de grains de sécrétion excrétés sur le nombre cumulé de tubes excrétés² lors de la simulation 7. De la même manière, les Fig. 6.8(c) et 6.8(d) illustrent la simulation 8. Les paramètres des simulations 7 et 8 sont présentés dans le Tab. 6.4.

Lors de la simulation 7, le diamètre des tubes est de $D_{tub} \simeq 30 \text{ nm}$ et leur longueur est de $L_{tub} \simeq 200 \text{ nm}$ (voir Tab. 6.4). Calculons le volume V_{tub} , la surface S_{tub} et le rapport de volume sur surface R_{tub} des tubes :

$$\begin{aligned} V_{tub} &\simeq \pi \times (D_{tub}/2)^2 \times L_{tub} \simeq 141 \times 10^3 \text{ nm}^3 \\ S_{tub} &\simeq 2 \times \pi \times (D_{tub}/2) \times L_{tub} + 2 \times \pi \times (D_{tub}/2)^2 \simeq 20 \times 10^3 \text{ nm}^2 \\ R_{tub} &\simeq V_{tub}/S_{tub} \simeq 7 \text{ nm} \end{aligned}$$

2. Le choix de l'illustration d'une moyenne est lié à des choix faits lors de l'implantation.

Simulation	Paramètre	Valeur
Simulation 7	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Sacculé 1</i> à chaque itération	10
	Surface par activateur	360000 nm^2
	Diamètre des tubes	30 nm
	Longueur des tubes	200 nm
Simulation 8	Volume initial des saccules	0
	Surface initiale des saccules	0
	Vésicules transportées vers le <i>Sacculé 1</i> à chaque itération	10
	Surface par activateur	360000 nm^2
	Diamètre des tubes	30 nm
	Longueur moyenne des tubes	300 nm

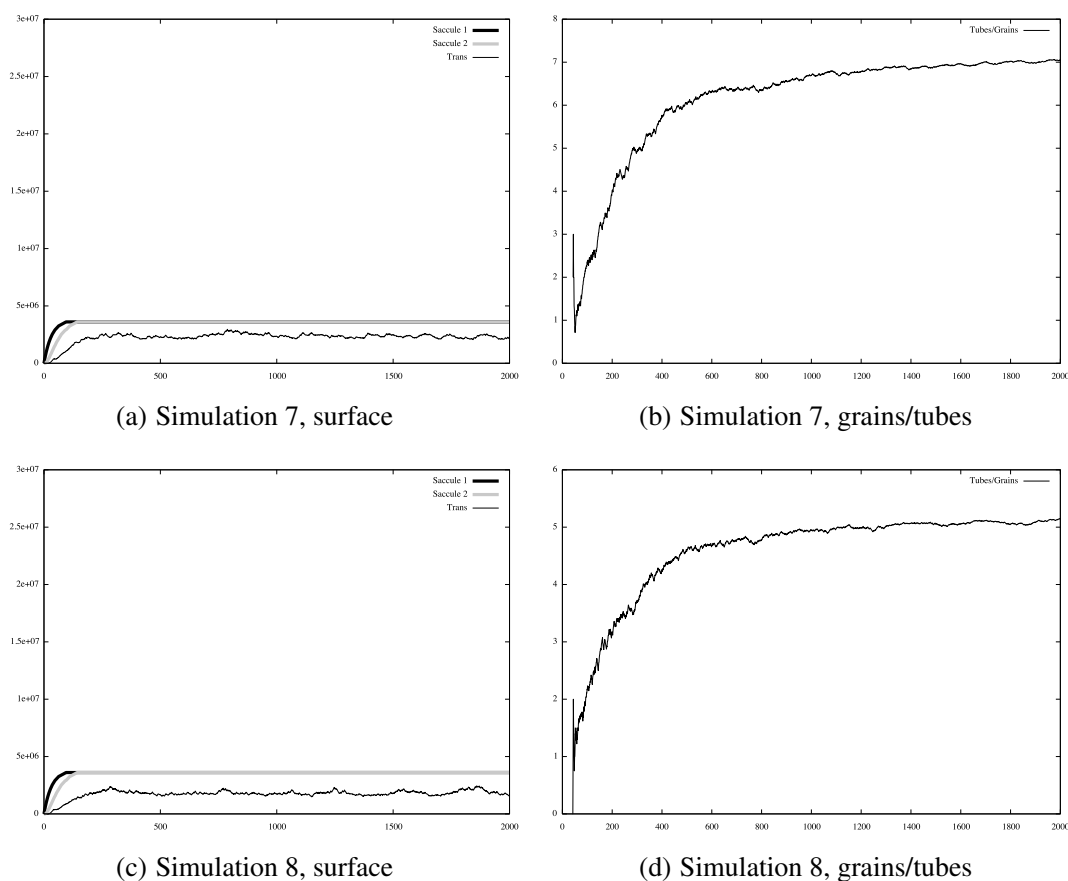
Tab 6.4 – Paramétrage de deux simulations avec création de tubes en *trans*

FIGURE 6.8 – Résultat des simulations 7 et 8

On remarque sur la Fig. 6.8(a) que ce paramétrage des tubes permet d'atteindre un état stationnaire dans la région *trans*. En effet, la surface du saccule *Trans* converge à partir de 200 itérations. Étudions cet équilibre en analysant la courbe de la Fig. 6.8(b). On remarque que le rapport du nombre cumulé de grains de sécrétion excrétés sur le nombre cumulé de tubes excrétés converge vers 7. Intuitivement, en moyenne, 7 tubes sont produits en *trans* pour un grain de sécrétion produit. Établissons par calcul

n_{tub} , le nombre de tubes nécessaires – par grain de sécrétion – afin d'atteindre l'équilibre. Le calcul est analogue au calcul de n_{ves} , le nombre de petites vésicules nécessaires afin d'équilibrer les rapports des flux entrants et sortants du saccule *Trans*. Dans le calcul suivant, V_{gra} et S_{gra} sont respectivement le volume et la surface d'un grain de sécrétion :

$$\begin{aligned} \frac{V_{gra} + n_{tub} \times V_{tub}}{S_{gra} + n_{tub} \times S_{tub}} &= 10 \\ n_{tub} &= \frac{10 \times S_{gra} - V_{gra}}{V_{tub} - 10 \times S_{tub}} \\ n_{tub} &\simeq 7,6 \end{aligned}$$

Ainsi, les entrées et sorties du saccule *Trans* sont équilibrées lorsqu'environ 7,6 tubes sont créés pour chaque grain de sécrétion. Ceci explique la convergence du rapport entre le nombre cumulé de grains de sécrétion et de tubes excrétés. Notons que puisque la courbe de la Fig. 6.8(b) représente l'évolution d'un rapport de quantités cumulées, elle converge plus lentement que la courbe de la Fig. 6.8(a). Notons enfin qu'à nouveau, les volumes des saccules évoluent de la même manière que dans les simulations précédentes. D'après les experts biologistes, un rapport d'environ 7 entre le nombre de grains de sécrétion et de tubes créés paraît réaliste et corrobore les observations. Remarquons que comme la longueur des tubes est grande comparée à l'épaisseur d'une coupe, le raffinement de ce rapport à l'aide d'un modèle probabiliste – tel que celui utilisé précédemment – n'est pas pertinent.

Dans la simulation 8, les tubes ont 30 nm de diamètre et 200 nm de longueur. Par des calculs analogues à ceux effectués dans le cadre de la simulation 7, on obtient que leur rapport de volume sur surface est d'environ 7.15. On remarque sur la courbe de la Fig. 6.8(c), qu'un tel paramétrage permet également d'équilibrer le saccule *Trans*. À présent, l'équilibre est atteint lorsqu'environ 5 tubes sont produits pour chaque grain de sécrétion (voir Fig. 6.8(d)). À nouveau, cet ordre de grandeur paraît réaliste.

Ainsi, les simulations 7 et 8 montrent que l'excrétion de tubes dans la région *trans*, et par conséquent la présence d'une zone tubulaire, permet d'atteindre un état d'équilibre. Contrairement au raffinement du modèle vésiculaire par dégradation de surface (voir section 6.3.2) ou par excrétion de petites vésicules, ce nouveau modèle apparaît réaliste. En effet, les tubes ne sont pas constitués d'une unique surface, mais également d'un volume englobé par cette surface. De plus, le nombre de tubes nécessaires pour équilibrer le saccule *Trans* est compatible avec les observations faites par les biologistes.

Ajout de volume et zones tubulaires

Dans cette section, nous proposons une autre manière d'équilibrer le saccule *Trans* à l'aide de tubes. Considérons des tubes de diamètre $D_{tub2} \simeq 40$ nm et de longueur $L_{tub2} \simeq 350$ nm. D'après les experts biologistes, de tels tubes sont réalistes : leur longueur est raisonnable et leur diamètre est inférieur à celui des vésicules de 60 nm. Calculons le volume V_{tub2} , la surface S_{tub2} et le rapport de volume sur surface R_{tub2} de tels tubes :

$$\begin{aligned} V_{tub2} &\simeq \pi \times (D_{tub2}/2)^2 \times L_{tub2} \simeq 439 \times 10^3 \text{ nm}^3 \\ S_{tub2} &\simeq 2 \times \pi \times (D_{tub2}/2) \times L_{tub2} + 2 \times \pi \times (D_{tub2}/2)^2 \simeq 46 \times 10^3 \text{ nm}^2 \\ R_{tub2} &\simeq V_{tub2}/S_{tub2} \simeq 9.5 \text{ nm} \end{aligned}$$

Deux éléments sont à noter, ces tubes présentent une surface $S_{tub2} \simeq 46 \times 10^3 \text{ nm}^2$ proche de la surface d'un grain de sécrétion $S_{gra} \simeq 45 \times 10^3 \text{ nm}^2$. De plus, le rapport $R_{tub2} \simeq 9.5 \text{ nm}$ de tubes est analogue au rapport $R_{ves} \simeq 10 \text{ nm}$ des vésicules. Ceci signifie que l'excrétion seule de tels tubes – nous oublions un instant les grains de sécrétion – ne crée pas de déséquilibre dans la région *trans*.

Si de tels tubes sont effectivement la seule structure à être excrétée depuis le saccule *trans*, se pose la question de l'origine des grains de sécrétion observés sur les micrographes issus de la microscopie électronique. D'après les biologistes, la formation des tubes dans la région *trans* du Golgi induit une concentration de matière dans certaines régions des tubes. Les sections de tubes concentrées – par exemple en protéines – excrétées du saccule *Trans* présentent alors un déficit en eau. Ainsi, afin de rétablir l'équilibre osmotique, de l'eau doit entrer dans la zone tubulaire. Notons que la perméabilité de certaines membranes aux molécules d'eau est un mécanisme connu en biologie. De cet ajout d'eau, c'est-à-dire de cette augmentation de volume, peut résulter la transformation d'une structure tubulaire en un grain de sécrétion. Ce modèle est illustré en Fig. 6.9. On remarque que comme dans la section précédente, deux types de structures bordent la région *trans* : des structures tubulaires et des grains de sécrétion. Ici encore, la présence des tubes est nécessaire pour la simulation d'un appareil de Golgi à l'état stationnaire.

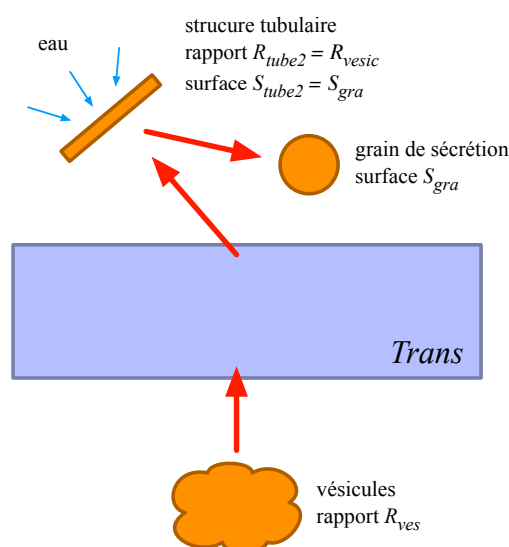


FIGURE 6.9 – Principe de l'ajout de volume dans les grains de sécrétion

6.3.4 Bilan

Notre étude des rapports de volume sur surface des compartiments biologiques de l'appareil de Golgi montre que le modèle vésiculaire entraîne une croissance critique de la membrane du saccule *Trans*. Ainsi, le modèle vésiculaire original – c'est-à-dire sans rééquilibrage *ad hoc* du ratio volume/surface du saccule *Trans* – n'est pas réaliste du point des systèmes dynamiques aux régimes stationnaires.

Nous avons vu également qu'une voie de dégradation de membranes pures (c'est-à-dire sans quantité de volume associée) permet d'observer, par simulation, un appareil de Golgi stationnaire. Cependant, une telle voie de dégradation n'est pas recevable du point de vue biologique. L'ajout d'un mécanisme d'excrétion de tubes depuis le saccule *Trans* permet d'obtenir des résultats proches des simulations par dégradation de membranes. De plus, ce mécanisme présente l'avantage de corroborer la présence de petits tubes parfois observés à la périphérie de la région *trans* du Golgi.

Ainsi le modèle vésiculaire peut être raffiné en un nouveau modèle où une zone *trans* tubulaire – dont l'existence est confirmée par l'observation – nous permet d'obtenir un appareil de Golgi dont la structure au régime stationnaire est stable.

6.4 Modèle de la maturation

Dans cette section, nous adaptons notre étude des rapports de surface sur volume dans le cas du modèle de la maturation. Commençons par rappeler brièvement les principes de base du modèle de la maturation, introduit dans la section 5.2.2. Dans ce modèle, le transport des protéines le long de l'appareil de Golgi est le fait d'un mouvement rétrograde de la pile de saccules ; on parle de maturation des saccules. Ce principe est illustré sur la Fig. 6.10, le mouvement de la pile est représenté par les flèches qui connectent les saccules. Comme nous l'avons vu dans le chapitre 5, le transport rétrograde des vésicules (représenté par les flèches orientées vers le bas) permet d'assurer le ravitaillement de la région *cis* en enzymes. Remarquons que comme dans la section précédente, la région *cis* n'est pas représentée : nous nous focalisons sur la partie médiane de l'appareil de Golgi. Enfin, ajoutons qu'à l'origine, ce modèle a été motivé par la nécessité de palier le fort coût en énergie des mécanismes d'emballage et de dépaquetage des vésicules responsables du transport des protéines dans le modèle vésiculaire. Cette hypothèse de minimisation du coût en énergie repose sur un nombre de vésicules significativement moindre par rapport au modèle vésiculaire.

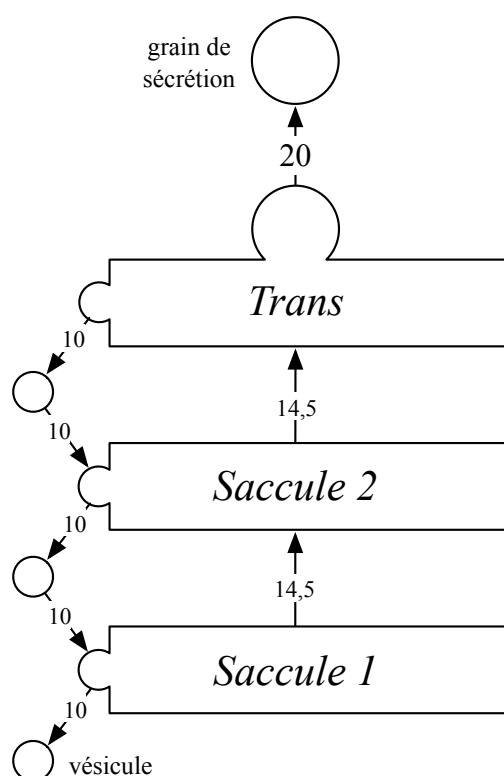


FIGURE 6.10 – Le modèle de la maturation

6.4.1 Un transport rétrograde coûteux

Déterminons à présent les rapports de volume sur surface des différents compartiments. En ce qui concerne les vésicules et grains de sécrétion, nous reprenons les valeurs utilisées lors de l'étude du modèle vésiculaire (voir Tab. 6.1, page 140). Dans le modèle de la maturation, nous considérons que les saccules, qui migrent le long de la pile, sont de forme cylindrique – proche d'une assiette – et apparaissent dans la région médiane avec un diamètre $D_{sac} \simeq 1500 \text{ nm}$ et une longueur (ou épaisseur) $L_{sac} \simeq 30 \text{ nm}$. Calculons le volume V_{sac} , la surface S_{sac} et le rapport de volume sur surface R_{sac} d'un saccule :

$$V_{sac} = \pi \times (D_{sac}/2)^2 \times L_{sac} \simeq 53014 \times 10^3 \text{ nm}^3$$

$$S_{sac} = 2 \times \pi \times (D_{sac}/2) \times L_{sac} + 2 \times \pi \times (D_{sac}/2)^2 \simeq 3675 \times 10^3 \text{ nm}^2$$

$$R_{sac} = V_{sac}/S_{sac} \simeq 14.5 \text{ nm}$$

Par hypothèse, dans le modèle de la maturation, chaque saccule se désagrège totalement lors de sa migration le long de l'appareil de Golgi. Le modèle de la maturation est illustré en Fig. 6.10. Comme précédemment, les rapports de volume sur surface des différents compartiments étiquettent les flèches. Comme le montre la figure, chaque saccule qui arrive dans la région *trans* est éclaté en deux types de compartiments : des vésicules rétrogrades et des grains de sécrétion. De plus, nous pouvons ignorer les mécanismes d'emballage et de dépaquetage des vésicules situées hors de la région *trans*. En effet, sur la figure, nous remarquons que les vésicules qui bourgeonnent à la surface du *Saccule 2* sont compensées par les vésicules transportées depuis le saccule *Trans*. Ainsi, nous pouvons considérer que le mécanisme de désaggrégation a lieu principalement dans la région *trans*.

Comme le rapport de volume sur surface d'un saccule est $R_{sac} \simeq 14.5$ nous savons que le rapport global de l'ensemble des vésicules et grains de sécrétion résultats du désaggrégement du saccule *Trans* est également de 14.5 nm . Calculons à présent le nombre n_{rves} de vésicules rétrogrades à produire par grain de sécrétion pour conserver ce rapport :

$$\begin{aligned} \frac{V_{gra} + n_{rves} \times V_{ves}}{S_{gra} + n_{rves} \times S_{ves}} &\simeq 14.5 \\ n_{rves} &\simeq \frac{14.5 \times S_{gra} - V_{gra}}{V_{ves} - 14.5 \times S_{ves}} \\ n_{rves} &\simeq 5 \end{aligned}$$

Il faut donc produire environ cinq vésicules rétrogrades par grain de sécrétion. Calculons à présent le volume $V_{n_{rves}}$ et la surface $S_{n_{rves}}$ de cinq vésicules :

$$V_{n_{rves}} = n_{rves} \times V_{ves} \simeq 550 \times 10^3 \text{ nm}^3$$

$$S_{n_{rves}} = n_{rves} \times S_{ves} \simeq 55 \times 10^3 \text{ nm}^2$$

Pour mettre en lumière l'efficacité du transport de la matière dans le modèle de la maturation, calculons le volume totale V_{tot} et la surface totale S_{tot} de cinq vésicules et un grain de sécrétion :

$$V_{tot} = V_{n_{rves}} + V_{gra} \simeq 1450 \times 10^3 \text{ nm}^3$$

$$S_{tot} = S_{n_{rves}} + S_{ves} \simeq 100 \times 10^2 \text{ nm}^2$$

Remarquons que le volume $V_{gra} \simeq 900 \times 10^3 \text{ nm}^3$ d'un grain de sécrétion représente environ 62 % de V_{tot} . Ainsi, si l'intégralité du saccule *Trans* est désaggrégé avec un rapport de cinq vésicules pour un grain de sécrétion, seulement 62 % du volume – c'est-à-dire de la matière soluble – du saccule *Trans* est effectivement excrété. Par un raisonnement analogue, puisque la surface $S_{gra} \simeq 45 \times 10^3 \text{ nm}^2$ d'un grain de sécrétion représente 45 % de S_{tot} , seulement 45 % de la surface – c'est-à-dire de la membrane – du saccule *Trans* est effectivement excrétée.

6.4.2 Bilan

En nous appuyant sur les estimations chiffrées de la section précédente, nous pouvons dire qu'il y a presque autant de matières transportées dans le sens rétrograde que de matière transportée dans le sens antérograde dans le modèle de la maturation. D'après les biologistes du groupe GolgiTop, ce bilan du transport de la matière va à l'encontre de l'idée originale du modèle de la maturation, censée optimiser le transport des protéines. Cependant, ce point est en débat dans la communauté biologique puisque certaines théories récentes prônent un équilibre entre le transport rétrograde et le transport antérograde dans l'appareil de Golgi ([Saraste 07]).

Remarquons qu'à nouveau, l'ajout de tubes au modèle de la maturation nous permet de compenser les différences de rapport de volume sur surface qui existent entre les grains de sécrétion antérogrades et les vésicules rétrogrades. Notamment, si l'on suppose – comme dans la section 6.3.3 – que les seules structures excrétées en *trans* sont des tubes de rapport 10 *nm*, les excrétions rétrogrades et antérogrades en *trans* s'équilibrent naturellement sans que soit mis en jeu une sécrétion coûteuse de nombreuses vésicules rétrogrades.

6.5 Modèle du flux continu de membranes

Le modèle du flux continu de membranes (voir section 5.2.3) ressemble, dans sa conception, au modèle de la maturation en ce sens que l'ensemble de la structure migre dans le sens antérograde. Une représentation schématique du modèle du flux continu de membranes est proposée en Fig. 6.11. Comme précédemment, nous représentons uniquement les parties médiane et *trans* de l'appareil de Golgi. Sur la figure, le mouvement de la structure est représenté par la flèche grisée, en fond. Comme nous l'avons vu dans le chapitre 5, du point de vue de la structure, le modèle du flux continu se distingue fortement des modèles discontinus – c'est-à-dire le modèle vésiculaire et le modèle de la maturation – par les tubes qui connectent les saccules. Dans ce modèle, les tubes sont situés à la périphérie de la pile de saccules. De plus, la perforation progressive des saccules dans le sens *cis* vers *trans* conduit à une région *trans* purement tubulaire. Retenons alors que le modèle du flux continu de membranes s'appuie sur une structure d'un seul tenant. Une conséquence directe de la connectivité des saccules est que les molécules – protéines, enzymes, *etc.* – peuvent se déplacer librement dans l'appareil de Golgi. Néanmoins, l'hypothèse naturelle est de supposer que celles-ci ont tendance à suivre le mouvement antérograde de la pile.

6.5.1 De l'insuffisance de l'étude des rapports de volume sur surface

Puisque le modèle du flux continu de membranes est constitué d'un unique compartiment – les grains de sécrétion sont extérieurs à l'appareil de Golgi – une discrimination basée sur l'étude des rapports de surface sur volume semble peu pertinente. Nous pouvons formuler l'observation suivante : si nous supposons qu'en moyenne la structure est stable – c'est-à-dire qu'il n'y a pas d'accumulation de matière dans l'appareil de Golgi – alors il existe un équilibre entre ce qui entre et ce qui sort. En particulier, le rapport de volume sur surface des structures qui entrent dans l'appareil de Golgi par la région *cis* est égal au rapport de volume sur surface des structures qui en sortent. Or, dans la section 5.2.3, nous avons vu que les grains de sécrétion peuvent prendre différentes formes (voir, Fig. 5.11, page 131). Prenons pour exemples les trois grains de sécrétion G , G' et G'' illustrés en Fig 6.12.

Le grain G est de diamètre $D_{gra} \simeq 120 \text{ nm}$, comme les grains considérés jusqu'à présent (voir section 6.3.1). Son volume est donc de $V_{gra} \simeq 900 \times 10^3 \text{ nm}^3$, sa surface est de $S_{gra} \simeq 45 \times 10^3 \text{ nm}^2$ et son rapport de volume sur surface est de $R_{gra} \simeq 20 \text{ nm}$.

Le grain G' présente une morphologie différente, il est composé d'une partie sphérique analogue au grain G et d'une partie tubulaire paramétrée comme les tubes de la simulation 7 (voir tableau Tab. 6.4,

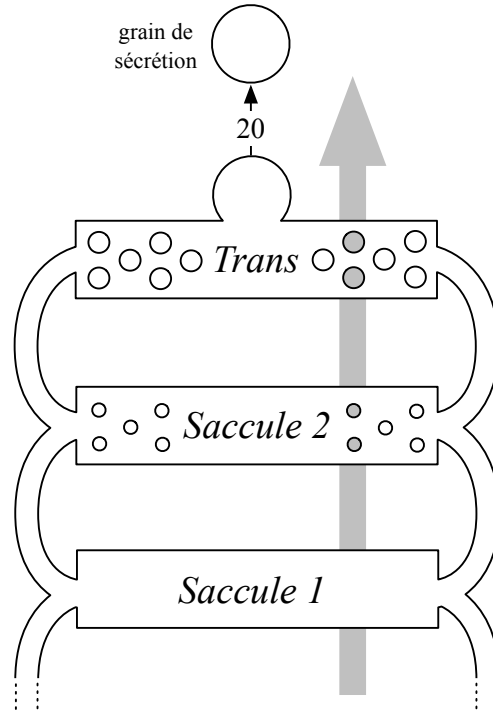


FIGURE 6.11 – Le modèle du flux continu de membranes

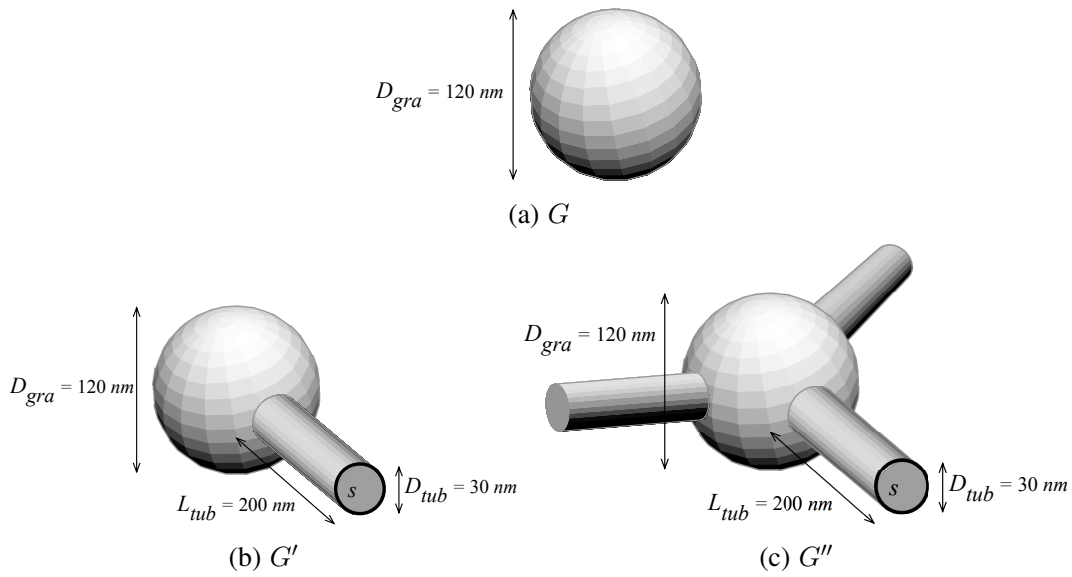


FIGURE 6.12 – Trois types de grains de sécrétion

page 151). Dans la section 5.2.3, nous avons vu que les mécanismes de pincement de la zone tubulaire *trans* expliquent les formes de ce type que peuvent prendre les grains de sécrétion. Calculons le volume V_{gra}' , la surface S_{gra}' et le rapport de volume sur surface R_{gra}' du grain G' . Rappelons que les tubes de la simulation 7 sont de volume $V_{tub} \simeq 141 \times 10^3 \text{ nm}^3$ et de surface $S_{tub} \simeq 20 \times 10^3 \text{ nm}^2$. Sur la Fig. 6.12, nous notons s les disques qui forment les bases des tubes et S_s leur surface :

$$V_{gra}' = V_{gra} + V_{tub} \simeq 1041 \times 10^3 \text{ nm}^3$$

$$S_{gra}' = S_{gra} + S_{tub} - 2 \times S_s \simeq 66 \times 10^3 \text{ nm}^2$$

$$R_{gra}' = V_{gra}' / S_{gra}' \simeq 15.5 \text{ nm}$$

Le grain G'' est composé d'une partie sphérique et de trois parties tubulaires également paramétrées comme les tubes de la simulation 7. Calculons son volume V_{gra}'' , sa surface S_{gra}'' et son rapport de volume sur surface R_{gra}'' :

$$V_{gra}'' \simeq V_{gra} + 2 \times V_{tub} \simeq 1182 \times 10^3 \text{ nm}^3$$

$$S_{gra}'' \simeq S_{gra} + 3 \times S_{tub} - (3 \times 2) \times S_s \simeq 96 \times 10^3 \text{ nm}^2$$

$$R_{gra}'' \simeq V_{gra}'' / S_{gra}'' \simeq 13.5 \text{ nm}$$

Les rapports $R_{gra} \simeq 20 \text{ nm}$, $R_{gra}' \simeq 15.5 \text{ nm}$ et $R_{gra}'' \simeq 13.5 \text{ nm}$ nous indiquent que pour les trois configurations considérées, les grains de sécrétion présentent des rapports de volume sur surface distincts. De plus, notons que la Fig. 6.12 ne présente pas une liste exhaustive des formes possibles que peuvent prendre les grains de sécrétion créés par pincement de la zone tubulaire *trans*. Intuitivement, on remarque que plus un grain de sécrétion est tubulaire, plus son rapport de volume sur surface est faible.

Ainsi, cette brève étude du modèle du flux continu de membranes nous indique que la variété des grains de sécrétion induit une variété des rapports de volume sur surface en sortie de l'appareil de Golgi. Par conséquent, le modèle du flux continu de membranes ne contraint pas la forme – ni les rapports – des structures qui entrent dans l'appareil de Golgi *via* la région *cis*. Ainsi, l'hypothèse d'un équilibre entre ce qui entre et ce qui sort de l'appareil de Golgi – et donc de l'égalité des rapports en entrée et en sortie – est nécessairement compatible avec le modèle du flux continu de membranes. La seule étude des rapports de volume sur surface n'est donc pas discriminante.

Dans les sections précédentes, nous avons raisonné sur des représentations purement abstraites des modèles de l'appareil de Golgi. Dans chacune de nos études, nous nous sommes donnés des compartiments abstraits – des saccules, des vésicules, des grains de sécrétion et des tubes – auxquels nous avons associé un volume et une surface. Cette modélisation simple permet déjà de raisonner sur les modèles étudiés. Cependant, dans la section précédente, nous avons vu que nos critères de discrimination basés sur l'étude des rapports de volume sur surface des compartiments ne permettent pas de valider – ou d'invalider – le modèle du flux continu de membranes.

Dans cette section, nous proposons une étude détaillée de la structure statique du modèle du flux continu de membranes. En particulier, nous montrons comment une représentation plus concrète de l'appareil de Golgi nous permet d'envisager de nouvelles pistes de discrimination. Dans les sections précédentes, les représentations graphiques d'appareil de Golgi (voir Fig. 6.3 page 139, Fig. 6.10 page 154 et Fig. 6.11 page 157) nous ont fourni pour seul support une manière intuitive de visualiser les différents compartiments ainsi que les règles de comportement mises en jeu dans les trois modèles.

Ici, nous proposons de raisonner directement sur des représentations graphiques, certes plus détaillées, de l'appareil de Golgi. Dans un premier temps, nous définissons une représentation tridimensionnelle pour chacun des trois modèles étudiés. Dans un second temps, nous en extrayons des coupes bidimensionnelles. La dernière étape de notre démarche est motivée par le besoin de validation. Les coupes construites sont alors comparées avec les images issues de l'observation.

6.5.2 Représentations tridimensionnelles des modèles de l'appareil de Golgi

Dans le chapitre 5, nous avons remarqué que le modèle vésiculaire et le modèle de la maturation présentent la même structure. Intuitivement, si l'on oublie les aspects dynamiques, les deux modèles sont identiques. Dans les deux cas, nous avons des saccules déconnectés munis de vésicules en leur périphérie et de grains de sécrétion aux abords de la région *trans*. Pour cette raison, dans cette section où nous nous focalisons sur la structure statique des trois modèles étudiés, nous ne distinguons pas le modèle vésiculaire et le modèle de la maturation. Pour désigner ces deux modèles, nous employons le terme *modèles discontinus* par opposition au modèle du flux *continu* de membranes.

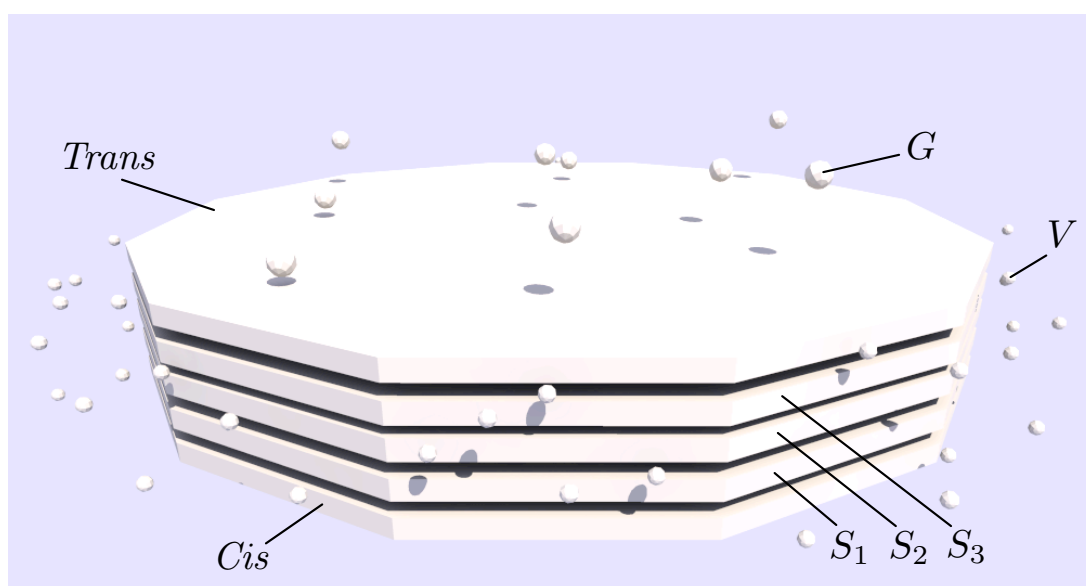
Du point de vue de la structure, nous savons que le modèle du flux continu de membranes se distingue principalement des modèles discontinus par les tubes qui connectent les saccules et par la perforation progressive des saccules dans le sens *cis* vers *trans*. Remarquons que ces caractéristiques propres au modèle du flux continu de membranes sont d'ordre topologique plutôt que géométrique. Intuitivement, nous nous intéressons à la présence ou à l'absence des tubes plutôt qu'à leur position précise ou leur forme. De fait, nous souhaitons proposer une représentation 3D des modèles de l'appareil de Golgi qui capture l'essence de leur structure tout en abstrayant au maximum leur géométrie. L'utilisation des modèles géométriques à base topologique prend alors tout son sens. Nous choisissons les cartes généralisées de dimension 3 comme support à nos modélisations.

Nos représentations 3D des modèles de l'appareil de Golgi sont proposées en Fig. 6.13³. La Fig. 6.13(a) propose une représentation 3D des modèles continus tandis que la Fig. 6.13(b) propose une représentation 3D du modèle du flux continu de membranes. Dans la suite, nous appelons nos représentations 3D respectivement : *pile d'assiettes* et *tour*. Notons que ces deux représentations sont le résultat de nombreuses discussions avec nos collègues biologistes. Ceux-ci ont analysé en détails chaque proposition intermédiaire en nous proposant régulièrement des mises à jour de la structure. En particulier, la tour constitue l'une des premières représentations tridimensionnelles du modèle du flux continu de membranes.

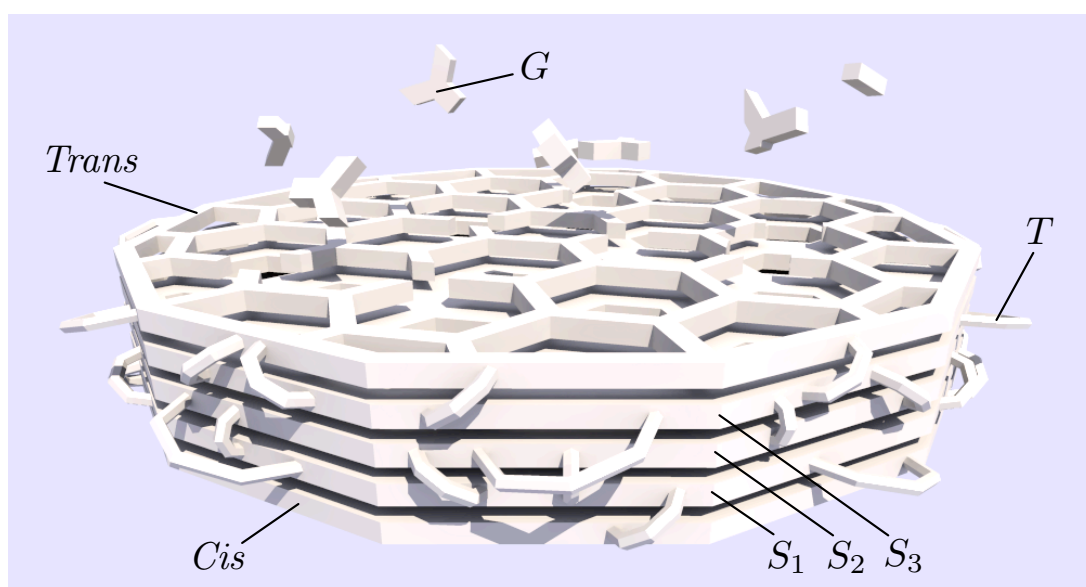
Dans chacun des deux modèles, nous avons choisi de représenter un saccule *cis* (représenté par un cylindre aplati noté *Cis*), un saccule *trans* (noté *Trans*) et trois saccules médians (notés S_1 , S_2 et S_3). Ce choix de cinq saccules au total s'explique par notre volonté d'approcher au maximum les observations faites en microscopie électronique. En ce sens, le micrographe de la Fig. 5.3 (page 124), où figurent cinq saccules, est représentatif des observations typiques. Enfin, comme précédemment, nous ne modélisons pas les structures situées entre le RE et l'appareil de Golgi.

On remarque qu'en dépit d'une géométrie simple, les différences de structure qui distinguent les modèles discontinus du modèle du flux continu de membranes apparaissent clairement. Dans la pile d'assiettes de la Fig. 6.13(a), les saccules sont déconnectés et des vésicules (représentées par des sphères, voir *V* sur la figure) sont situées en leur périphérie. À l'opposé, dans la tour de la Fig. 6.13(b), les saccules sont connectés par des tubes (représentés par des colonnes, voir *T* sur la figure). De plus, la tour présente une perforation progressive des saccules dans le sens *cis* vers *trans*, ce qui n'est pas le cas dans la pile d'assiettes. Bien qu'elles n'apparaissent pas sur la Fig. 6.13(b), le saccule S_2 présente déjà quelques perforations. Sur la figure, on remarque que ces perforations sont accentuées à la surface du saccule S_3 , tandis que le saccule *Trans* est presque totalement tubulaire. On remarque également que les grains de sécrétions, tous représentés par des sphères dans la pile d'assiettes (voir *G* sur la Fig. 6.13(a)), ont différentes structures dans le cas de la tour (voir *G* sur la Fig. 6.13(b)). Nous savons que ces variations s'expliquent par le mécanisme de formation des grains de sécrétion dans le modèle du flux continu de membranes (voir sections 5.2.3 et 6.5.1).

3. Nous avons construit ces objets à l'aide du modelleur Moka [Vidil 03], développé au laboratoire XLIM-SIC. Il s'agit d'un modelleur géométrique de 3-G-cartes. Moka propose un plongement polyédrique des objets, à l'origine de l'aspect anguleux des objets.



(a) La pile d'assiettes



(b) La tour

FIGURE 6.13 – Représentations 3D à base de 3-G-cartes des modèles de l'appareil de Golgi

6.5.3 Validation des modèles tridimensionnels par des coupes

Les Fig. 6.14(a) et 6.14(b) présentent respectivement une coupe de la pile d'assiettes et une coupe de la tour⁴. La première caractéristique remarquable est que dans chacune des coupes, les saccules ne présentent pas – ou peu – de connexions entre eux, ce qui est également le cas sur la plupart des coupes d'appareils de Golgi issues de la microscopie électronique (voir Fig. 5.3 page 124). De plus, dans les deux coupes, nous remarquons de petites structures notées *V'* situées à la périphérie des saccules. Sur les micrographes d'appareil de Golgi, de telles structures sont interprétées comme des vésicules. Sur la

4. Nous avons construit ces coupes en superposant – par transparence – de fines coupes de nos modèles tridimensionnels. De cette manière, nous obtenons un résultat visuellement proche des coupes réelles observées par microscopie électronique.

Fig. 6.14(a), ces structures sont le résultat de la coupe des vésicules qui bordent les saccules dans la pile d'assiettes (voir Fig. 6.13(a)). Remarquons que dans ce cas, elles apparaissent systématiquement sous la forme de disque. Sur la Fig. 6.14(b), ces structures ont des formes variées. Dans les deux cas, elles résultent de la coupe des tubes qui connectent les saccules de la tour (voir Fig. 6.13(b)). De la même manière, les coupes des grains de sécrétions – notées *G* sur les fig. 6.13(a) et 6.14(b) – présentent des formes irrégulières dans le cas de la tour.

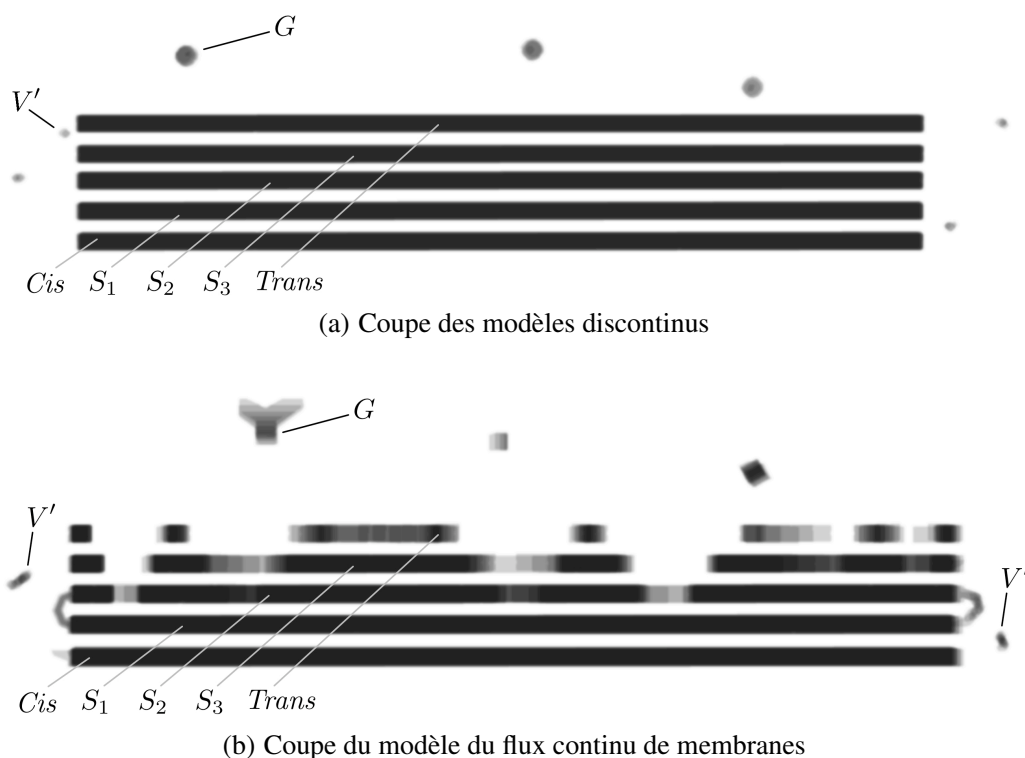


FIGURE 6.14 – Coupes des représentations 3D des modèles de l'appareil de Golgi

L'exemple de la Fig. 6.15 illustre la variété des formes qui apparaissent lorsque l'on coupe une sphère ou un cylindre, c'est-à-dire une vésicule ou un tube. Les Fig. 6.15(a), 6.15(b) et 6.15(c) sont associées respectivement aux plans de coupe des Fig. 6.15(d), 6.15(e) et 6.15(f). Sur la Fig. 6.15(a), une sphère est coupée par deux plans. Sur les deux plans de coupe apparaissent deux disques de taille différente, en raison du positionnement des plans de coupe. En Fig. 6.15(b), on coupe un cylindre horizontalement. Comme dans le cas de la coupe d'une sphère, on remarque qu'un disque apparaît sur le plan de coupe. Ceci suggère que ce que l'on interprète comme étant des vésicules sur les images prises au microscope électronique peut également être de petites sections de tubes. Enfin, en Fig. 6.15(c), on coupe un cylindre en biais. Dans ce cas, on remarque que la coupe prend une forme allongée, ce qui explique les formes allongées que prennent les coupes des tubes sur la Fig. 6.14(b).

Ainsi, bien que sur la plupart des micrographes de l'appareil de Golgi, on ne voit pas de tube en tant que tel, ce qui est usuellement interprété comme une fine section de vésicule peut également être interprété comme une petite section de tube. De plus, l'exemple de la Fig. 6.15 montre que seules ces coupes de tubes permettent d'expliquer les vésicules et grains de sécrétion allongés parfois observés sur les micrographes d'appareil de Golgi.

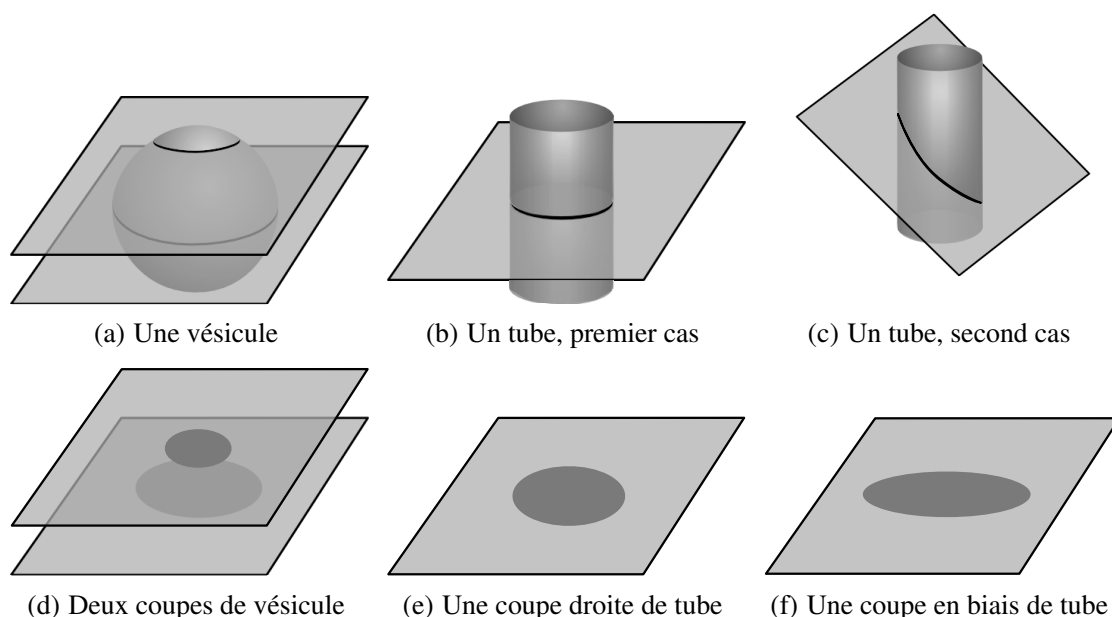


FIGURE 6.15 – Coupes de vésicules et de tubes

Un autre élément est à noter. Remarquons les coupes des saccules *Cis* et S_1 sont identiques dans les deux coupes. En revanche, dans la coupe de la tour, les saccules S_2 , S_3 et *Trans* apparaissent crénelés tandis qu'ils sont uniformes dans la coupe de la pile d'assiettes. Ces irrégularités sont corrélées à la perforation progressive des saccules. Une observation analogue peut être effectuée sur le micrographe de la page 124. En effet, on remarque que certains saccules présentent un aspect irrégulier. Ils sont composés d'une alternance de régions sombres et de régions plus claires. À nouveau, seule la tour présente des artefacts analogues.

6.5.4 Bilan

Dans cette section, à partir de représentations simples où l'on se focalise sur la topologie au détriment de la géométrie, nous avons montré que l'hypothèse d'une pile de saccules connectés – qui conceptuellement s'éloigne sensiblement des modèles discontinus traditionnels – est finalement plus proche de l'observation que l'hypothèse d'une structure déconnectée. Ainsi, le modèle du flux continu de membranes explique simplement la discontinuité des saccules ainsi que la variété des vésicules et grains de sécrétion observés sur les micrographes. De plus, la coupe de la tour nous montre que les micrographes où les saccules apparaissent déconnectés ne sont pas pour autant nécessairement dénués de tubes, ni de perforations.

6.6 Conclusion de notre étude

Dans ce chapitre, nous avons introduit notre méthode de discrimination de modèles biologiques. Nous avons pris pour exemple les trois modèles de l'appareil de Golgi introduits au chapitre 5 : les modèles vésiculaires et de la maturation, que nous qualifions de modèles continus, et le modèle du flux continu de membranes. Notre méthode est basée sur des simulations successives des trois modèles. Après chaque simulation, les phénomènes émergents (de la simulation) sont comparés avec l'observation biologique. Enfin, à l'issue de cette comparaison, les modèles sont mis à jours. Les modèles qui ne peuvent pas être mis à jour sont écartés.

Nous avons remarqué que le choix des données sur lesquelles portent la discrimination est une étape fondamentale. De fait, une part importante de ce chapitre est consacrée à l'étude de ces données. Dans un premier temps, nous nous sommes focalisés sur l'étude des rapports de volume sur surface des différents compartiments biologiques qui constituent l'appareil de Golgi. Nous avons vu que cette donnée simple permet déjà d'invalider le modèle vésiculaire et de mettre en cause le modèle de la maturation. Nous avons enfin conclu que l'étude de cette seule donnée n'est pas suffisante dans le cas du modèle du flux continu de membranes. Pour ce dernier modèle, nous avons proposé une première validation à partir de coupes de modèles 3D. Cette étude montre que le modèle du flux continu de membranes paraît plus réaliste que les modèles continus : il correspond mieux à l'observation.

Dans le chapitre suivant, nous souhaitons poser les bases d'un simulateur qui permette d'animer des structures topologiques 3D afin d'aller plus loin dans le processus de discrimination. Remarquons que nous avons considéré la pile d'assiettes ainsi que la tour d'un point de vue purement statique. Dans ce nouveau chapitre, nous montrons comment les règles de transformation de graphes introduites dans la partie I permettent, dans un simulateur, de prendre en considération les aspects dynamiques. En particulier, un tel simulateur doit permettre d'observer en détail les flux de matières à l'intérieur des compartiments biologiques. Dans le cas de l'appareil de Golgi, nous souhaitons par exemple observer les flux de protéines. Une telle modélisation, plus fine, permet d'envisager de nouveaux critères de discrimination.

PERSPECTIVES : VERS UN SIMULATEUR BIOLOGIQUE À BASE TOPOLOGIQUE

Sommaire

7.1	La simulation des compartimentations biologiques	166
7.1.1	Simulation et modèle	166
7.1.2	Quelques approches pour la simulation des compartimentations biologiques . .	167
7.2	Vers un simulateur biologique à base topologique	172
7.2.1	Deux modèles biologiques	172
7.2.2	Le noyau de 3-G-cartes	174
7.2.2.1	Notre structure de brins	174
7.2.2.2	Les opérations élémentaires du noyau	176
7.2.3	Plongements et sorties graphiques	180
7.2.3.1	Plongement	180
7.2.3.2	Les sorties graphiques	182
7.2.4	L'implantation des règles de transformation	185
7.2.4.1	Structure de données pour les règles de transformations	185
7.2.4.2	Application des règles de transformation	190
7.2.5	Bilan et perspectives	197

Dans ce chapitre, nous posons les bases de notre *simulateur biologique à base topologique*. Il s'agit avant tout d'un simulateur de modèles topologiques qui s'appuie sur les règles de transformation de graphes introduites dans la partie I. Nous le qualifions de « biologique » car nous le spécialisons pour la simulation de modèles biologiques comme ceux présentés dans le chapitre 6. De fait, il intègre des primitives dédiées à la manipulation de compartiments biologiques. Le cahier des charges du simulateur que nous proposons s'appuie sur le développement de notre premier prototype de simulateur dédié à la biologie.

Ce premier prototype comprend notamment une structure de données implantant les 3-G-cartes plongées géométriquement aux sommets, ainsi que quelques fonctions élémentaires de manipulation de cette structure. Il comprend également une structure de données permettant d'encoder des règles de transformation de graphes classiques – c'est-à-dire sans variable – ainsi qu'un moteur d'application de ces règles sur une 3-G-carte. Remarquons que ce premier prototype ne comporte pas d'interface graphique permettant de construire la 3-G-carte que l'on souhaite simuler. En revanche, un interfaçage avec le modèleur Moka [Vidil 03] (déjà basé sur des 3-G- cartes), permet d'exporter les objets créés avec ce dernier dans un format lisible par le simulateur. De plus, de manière à visualiser les simulations créées à l'aide de notre prototype, nous avons développé plusieurs sorties graphiques dont une en 3D permettant de suivre l'évolution des objets dans l'espace, et une seconde permettant de tracer des graphiques qui décrivent l'évolution des quantités de molécules dans les compartiments. Remarquons que notre prototype est composé d'environ 2000 lignes de code écrites en Caml [Oca]. Par conséquent, les différents codes sources proposés dans ce chapitre sont des codes simplifiés écrits en langage Caml et directement issus du prototype. Notons enfin que tout au long de ce chapitre, nous veillons à distinguer les caractéristiques de l'implantation existante de celles de ses futures extensions.

Dans la section 7.1, nous évoquons des travaux où les auteurs se sont intéressés à la modélisation et à la simulation en biologie. Nous nous focalisons sur les travaux dans lesquels on porte une attention particulière à la représentation des compartimentations biologiques. Ensuite, en section 7.2, nous présentons notre cahier des charges de simulateur.

7.1 La simulation des compartimentations biologiques

7.1.1 Simulation et modèle

Les *simulations informatiques* – ou plus simplement *simulations* – reproduisent par ordinateur un phénomène réel observable. Elles permettent de prévoir des comportements à partir d'un modèle. Dans la pratique, elles sont utilisées lorsque les méthodes analytiques ne permettent pas de calculer ce comportement. De fait, les simulations trouvent principalement leur intérêt lorsque l'on étudie un modèle complexe dont les équations sont difficiles à résoudre voire à établir. En général, les modèles complexes sont des modèles dans lesquels de nombreuses entités sont en interaction. Nous pouvons prendre pour exemple un compartiment biologique qui contient un ensemble de molécules dont les possibilités d'assemblage sont multiples, mais contribuent à former les complexes moléculaires observés.

Les simulations informatiques touchent de nombreux domaines d'applications dans les sciences expérimentales : physique, chimie, biologie, mais aussi économie, sociologie ou psychologie. Par exemple, en acoustique, la combinaison de techniques de modélisation 3D et de simulation permet d'étudier la manière dont le son est propagé dans la maquette virtuelle d'un amphithéâtre. De même, en psychologie, des simulations à base d'agents autonomes permettent d'exhiber certains comportements sociaux à partir d'un ensemble d'individus. On retrouve ici la notion de *phénomène émergent* qui, rappelons-le, désigne l'ensemble des comportements globaux que l'on peut exhiber à partir d'interactions définies localement entre les entités (ici des individus).

Lorsque l'on étudie un phénomène réel par simulation, le choix du modèle est une étape fondamentale. Souvent, les hypothèses de départ permettent de s'orienter vers le choix du modèle le mieux adapté. Prenons l'exemple des techniques de calcul de l'éclairement en synthèse d'images. Dans le cas d'une scène composée principalement de surfaces mates – comme un bois non laqué – les modèles d'éclairement globaux tels que la radiosité [Cohen 93] sont préconisés. En revanche, si l'on suppose que la plupart des surfaces sont brillantes, les modèles à base de tracé de photons [Jensen 01] – basés sur des interactions locales – sont mieux adaptés.

Le choix du modèle dépend également de l'échelle d'observation du phénomène étudié. Prenons l'exemple de l'évolution des phénotypes¹ dans une famille d'individu. Dans ce cas, la modélisation des gènes à grande échelle par les caractéristiques qu'ils déterminent – on parle notamment du « gène de la couleur des yeux » – est suffisante pour effectuer des études statistiques sur une population. Une représentation à plus moyenne échelle de l'ensemble du génotype² ainsi que des interactions connues entre les gènes permet une étude plus précise des caractéristiques héréditaires. Enfin, à petite échelle, une modélisation des protéines codées par les gènes ainsi que des réactions chimiques sous-jacentes rend difficile l'étude des phénomènes émergents à l'échelle du phénotype. En effet, l'échelle des protéines permet de rendre compte de l'origine du phénotype mais fournit un modèle complexe sur le plan numérique.

Enfin, dans le chapitre 5, trois modèles de même échelle capturent les mécanismes à l'origine du transport des protéines dans l'appareil de Golgi. Ce sont les modèles vésiculaire, de la maturation et du flux continu de membranes. Dans ce cas, la simulation permet de comparer les phénomènes qui émergent de chacun de ces modèles ; c'est l'objet de la méthodologie de discrimination que nous présentons à la section 6.2.

7.1.2 Quelques approches pour la simulation des compartimentations biologiques

La biologie est un lieu privilégié pour l'étude de phénomènes complexes. En effet, de nombreux mécanismes biologiques – en particulier au sein de la cellule – sont le fait d'interactions entre de nombreuses molécules, souvent situées dans des compartiments eux-mêmes en interaction. La voie sécrétoire de la cellule, et donc l'appareil de Golgi, s'inscrit notamment dans ce cadre. Les mathématiques sont un outil essentiel pour l'étude de certains de ces mécanismes. En particulier les systèmes à base d'équations différentielles sont souvent utilisés pour modéliser l'évolution des concentrations [Murray 08a, Murray 08b] dans des compartiments. La simulation informatique intervient lorsque les méthodes de résolution classique ne permettent pas d'établir les solutions d'un système d'équations. Notamment, la méthode des éléments finis [Allaire 07, Zienkiewicz 05] est l'une des techniques de simulation numérique qui permettent d'approcher les solutions d'un système à base d'équations différentielles.

Les modélisations par équations permettent de raisonner globalement sur un système. Il ne s'agit pas de manipuler localement les entités du système étudié, mais d'en exhiber des comportements globaux dans lesquels sont mis en jeu des ensembles d'entités. Ce sont ces comportements qui sont mis en équation dans les modèles. Bien souvent, les paramètres d'une équation sont liés à plusieurs entités du système étudié. Par exemple, on manipule une concentration en protéines plutôt que les protéines elles-mêmes. Cette caractéristique des modélisations à base d'équations permet d'abstraire de nombreuses entités, ce qui est essentiel devant le gigantisme des populations – en molécules, cellules ou même individus – manipulées en biologie. Remarquons que les phénomènes qui émergent des simulations à base d'équations différentielles sont d'autant plus pertinents que la taille des populations manipulées est élevée.

Cependant, le fait de raisonner au niveau global – c'est-à-dire à assez grande échelle devant l'échelle des molécules – rend difficile la mise en équations lorsque les interactions possibles entre les entités (notamment les différentes réactions biochimiques) sont elles-mêmes nombreuses. De même, lors de la mise

1. Le phénotype est l'ensemble des traits observables (par exemple la couleur ou la forme des yeux) qui caractérisent un être vivant.

2. Le génotype est l'ensemble des gènes d'un individu.

au point du modèle, la modification locale des comportements – comme l’ajout d’une nouvelle réaction – implique souvent la refonte d’une grande partie du système d’équations.

Ces deux principales limites des modélisations à base de systèmes d’équations et des simulations numériques sous-jacentes ont conduit les biologistes à s’intéresser à des outils centrés sur une description locale des comportements. Parmi ceux-ci, les modélisations à base de règles de comportement jouent un rôle essentiel. Ici, les modèles simulés sont vus comme un état initial muni d’un ensemble de règles locales de comportement. Remarquons que c’est sous cet angle que nous avons évoqué le modèle vésiculaire dans la section 6.3. Dans ce cas, la simulation nous permet de faire évoluer l’état initial, par application des règles de comportement.

Parmi les nombreuses modélisations à base de règles utilisées en biologie, nous nous focalisons sur celles où les phénomènes biologiques sont vus comme des *systèmes dynamiques à structure dynamique* (ou $(SD)^2$) [Spicher 06]. Intuitivement, dans de tels systèmes, ce sont de concert les structures et les entités qui évoluent au cours du temps. La mitose cellulaire est un exemple de $(SD)^2$. Lors de la mitose, une cellule est divisée en deux. De fait, la cellule étant constituée de nombreux compartiments, la mitose induit une modification forte de la compartimentation – et donc de la structure – d’une cellule. De plus, la mitose entraîne un partage du matériel génétique ainsi que des molécules qui habitent la cellule ; les entités sont elles-mêmes dynamiques.

Les différents modèles de l’appareil de Golgi introduits dans la section 5.2 s’inscrivent dans le cadre des $(SD)^2$. Dans chacun des modèles, la structure est dynamique. Par exemple, dans le modèle vésiculaire, des vésicules bourgeonnent depuis un saccule et fusionnent avec les saccules supérieurs. Dans le modèle de la maturation, les saccules sont également en mouvement. Enfin, dans l’hypothèse du flux continu de membranes, les saccules sont perforés progressivement en une zone tubulaire. De plus, dans chacun des modèles, les entités – c’est-à-dire les protéines – sont véhiculées d’un compartiment à un autre et sont parallèlement transformées par des enzymes.

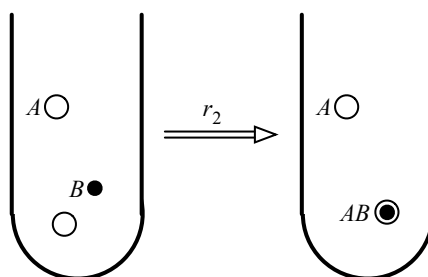
Dans la suite de cette section, nous évoquons différentes approches à base de règles dans le contexte de la biologie. Nous les distinguons selon qu’elles permettent ou non de prendre en compte la dynamique des structures, et avec quelle précision.

L’approche par « vase clos »

Dans les modélisations dites par « vase clos », on considère que toutes les molécules baignent dans un même compartiment. On peut prendre pour exemple la métaphore du tube à essai dans lequel potentiellement, les molécules de la solution chimique peuvent toutes interagir entre elles. Le modèle des multi-ensembles – utilisé notamment en chimie artificielle [Dittrich 01] – permet de modéliser, puis de simuler, de tels compartiments. Intuitivement, un multi-ensemble [Syropoulos 01] est un ensemble mathématique avec redondance. Il s’agit d’une structure informatique classique où à chaque élément est associé un nombre d’occurrences.

Considérons trois espèces moléculaires : A , B et AB , cette dernière étant le résultat de la complexation de deux molécules A et B . Considérons à présent un compartiment comportant deux molécules A et une molécule B . On modélise ce compartiment par le multi-ensemble $\{A, A, B\}$. Une représentation graphique de $\{A, A, B\}$ est donnée dans la partie gauche de la Fig. 7.1. On se donne alors un ensemble de réactions chimiques, représentées par des règles de réécriture. Elle permettent de faire évoluer le système par simulation. Dans notre cas, prenons deux règles :



FIGURE 7.1 – Application de la règle r_2 sur un multi-ensemble $\{A, A, B\}$

La règle r_1 nous indique que deux molécules A réagissent et synthétisent une nouvelle molécule B . La seconde règle, r_2 , décrit la complexation de deux molécules A et B en une nouvelle molécule AB . On remarque que suite à l'application de r_2 , les molécules A et B disparaissent.

Afin de rendre la simulation plus réaliste, il est important de considérer que comme dans la réalité, toutes les réactions chimiques possibles peuvent se produire simultanément. Ainsi, intuitivement, une simulation peut être vue comme une boucle d'application de règles dans laquelle à chaque itération, toutes les règles filtrées s'appliquent. Dans l'exemple, au début de la simulation, le compartiment est modélisé par le multi-ensemble $\{A, A, B\}$. Il s'agit de l'état initial du système. Lors de la première boucle de simulation, les deux règles r_1 et r_2 peuvent être appliquées ; le système contient deux molécules A et une molécule B , les motifs gauches de r_1 et r_2 peuvent donc l'un comme l'autre être filtrés. Cependant, les deux règles ne peuvent pas être appliquées simultanément. Il est alors nécessaire de choisir quelle règle sera effectivement appliquée. Dans la pratique, on attribue généralement une priorité à chaque règle de transformation. Dans l'exemple, en attribuant la priorité la plus élevée à la règle r_2 , le multi-ensemble $\{A, A, B\}$ est réécrit en $\{A, AB\}$ (représenté graphiquement dans la partie droite de la Fig. 7.1) lors de la première boucle de simulation.

Cet exemple simple met en exergue une notion importante : la notion de *stratégie d'application des règles*, déjà évoquée en section 6.2.1. Elle détermine dans quel ordre sont appliquées les règles au cours d'une boucle de simulation. Retenons qu'il existe principalement deux types de stratégies d'application : les *stratégies déterministes* et les *stratégies stochastiques*. Dans le cas des stratégies déterministes, sous les mêmes conditions – c'est-à-dire pour un même état initial, un même jeu de règles et une même stratégie déterministe – deux simulations donnent toujours le même résultat. À l'opposé, dans le cas des stratégies stochastiques, deux simulations soumises aux mêmes conditions initiales peuvent produire des résultats différents.

Remarquons que les modélisations par « vase clos » ne permettent pas de capturer la dynamique des compartiments, pourtant essentielle pour une grande partie des mécanismes biologiques. Ici, on considère que les molécules baignent toutes dans un même compartiment. Dès lors, les modélisations par « vase clos » ne s'inscrivent pas dans la modélisation des $(SD)^2$. Cependant, les modélisations à base de multi-ensembles sont à l'origine de plusieurs extensions où l'on sait capturer une certaine dynamique des compartiments. Dans de ce manuscrit, nous les appelons les *langages biochimiques*.

Les langages biochimiques

À l'image des modélisations à base de multi-ensembles, le langage BIOCHAM [Calzone 06] s'appuie sur un formalisme simple de réactions biochimiques. Il est ainsi simple à mettre en œuvre par les biologistes puisque proche des notations chimiques habituelles. Du point de vue informatique, BIOCHAM présente l'avantage d'être bien adapté aux techniques d'analyse par model checking. Ces techniques, qui

sortent du strict cadre de ce manuscrit, permettent de vérifier qu'une modélisation satisfait les propriétés du système étudié. Enfin, contrairement aux modélisations par « vase clos », le langage BIOCHAM permet d'indiquer dans quels compartiments sont situées les molécules modélisées. Cette extension ne permet pas encore de modifier la structure de la compartimentation, mais permet néanmoins d'écrire des règles qui transportent des molécules d'un compartiment à un autre. Prenons pour exemple le jeu de règles suivant :

$$\begin{aligned} r_1 : & A :: \text{noy} \rightarrow A :: \text{cyto} \\ r_2 : & A :: \text{cyto} + B :: \text{cyto} \rightarrow AB :: \text{cyto} \end{aligned}$$

La règle r_1 transporte une protéine A depuis le noyau d'une cellule vers le cytoplasme. Le nom d'une protéine est suivi de $:: \text{noy}$ ou $:: \text{cyto}$ selon qu'elle se situe respectivement dans le noyau ou dans le cytoplasme. On remarque qu'un nommage astucieux des protéines permet de modéliser le transport. La règle r_2 indique que lorsque deux protéines A et B sont dans le cytoplasme, elles se complexent en une nouvelle protéine AB .

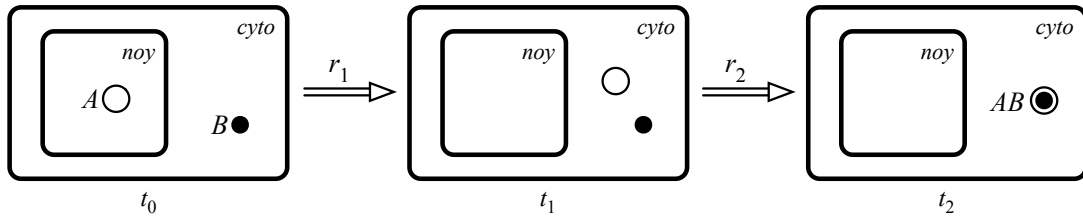


FIGURE 7.2 – Transport des protéines dans BIOCHAM

Le résultat d'une simulation simple qui s'appuie sur ces deux règles est illustré en Fig. 7.2. L'état initial du système – c'est-à-dire au temps t_0 – est représenté à gauche sur la figure. On remarque qu'au départ, le noyau est muni d'une protéine A . De même, une protéine B est située dans le cytoplasme. Lors de la première boucle de simulation, seule la règle r_1 peut être appliquée. Ainsi, au temps t_1 , les deux protéines sont dans le cytoplasme. Lors de la seconde itération, seule la règle r_2 peut être appliquée. En fin de simulation, le système comprend un unique complexe AB dans le cytoplasme.

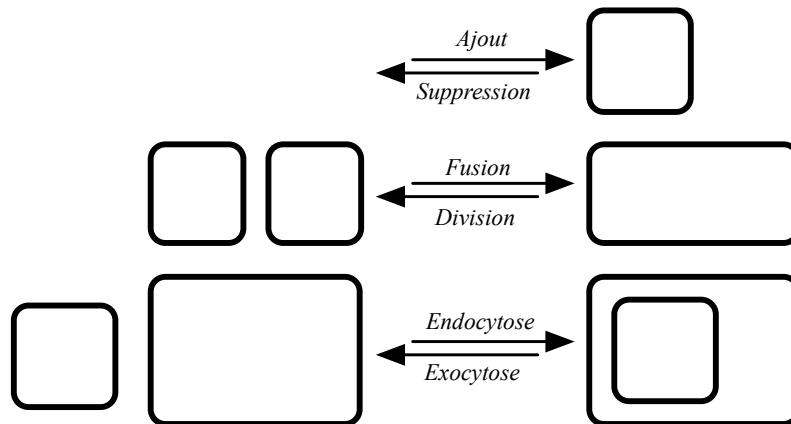


FIGURE 7.3 – La dynamique des compartiments dans BioAmbients et Brane Calculi

Les langages biochimiques BioAmbients [Regev 04] et Brane Calculi [Cardelli 05], tous deux inspirés du langage de processus π -calcul [Milner 92], permettent de modéliser une certaine dynamique

des compartiments. De fait, ils s'inscrivent dans le cadre de la modélisation des $(SD)^2$ biologiques. La Fig. 7.3 illustre les modifications des compartiments que l'on sait représenter dans ces deux langages. Les règles d'*ajout*, de *suppression*, de *fusion* et de *division* sont intuitives. La règle d'*endocytose* permet de déplacer un compartiment dans un autre. Il s'agit d'un mécanisme biologique classique notamment mis en oeuvre dans le transport de molécules vers l'intérieur de la cellule. À l'inverse, l'*exocytose* – également d'inspiration biologique – permet d'évacuer un compartiment depuis son compartiment d'origine vers l'extérieur. Illustrons le langage BioAmbients à l'aide d'une règle de fusion de compartiment inspirée des mécanismes mis en oeuvre dans l'appareil de Golgi :

$$r : \text{vesic}[fusion + n, A] \mid \text{sacc}[merge - n, B] \rightarrow \text{sacc}[A \mid B]$$

La règle r modélise la fusion d'une vésicule avec un saccule. Une représentation graphique de r est donnée en Fig. 7.4. La syntaxe utilisée dans r est proche de la syntaxe du langage π -calcul, où des processus communiquent entre eux par envoi de messages. Dans la partie gauche de la règle r , le terme $\text{vesic}[fusion + n, A]$ indique notamment que la vésicule contient une protéine A . De plus, $fusion + n$ signifie que la vésicule « envoie une demande de fusion dans un canal de communication n ». Le saccule, identifié par $\text{sacc}[merge - n, B]$ dans la partie gauche de r , contient une protéine B . Le terme $merge - n$ signifie que le saccule « accepte les demandes de fusion transmises *via* le canal n ». Dans la partie droite de la règle, le saccule – identifié par $\text{sacc}[A \mid B]$ – contient sa protéine d'origine B et la protéine A issue de la vésicule.

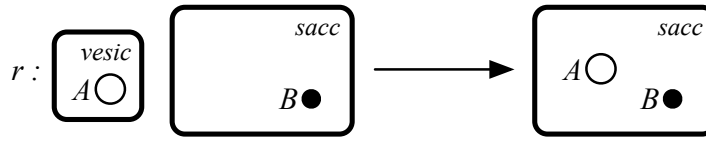


FIGURE 7.4 – La fusion d'une vésicule et d'un saccule dans BioAmbients

Les langages biologique BioAmbients et Brane Calculi permettent de manipuler les compartiments biologiques de manière totalement abstraite. Se positionnant dans le sillage du π -calcul, ils bénéficient des résultats ainsi que du support de toute une communauté. Cependant, le prix de ce savoir-faire est une certaine difficulté à prendre en compte des mécanismes compartimentaux plus complexes. En effet, certains des mécanismes topologiques mis en jeu en biologie ne peuvent être modélisés sans une représentation concrète des compartiments où les différents éléments de la structure sont reliés à des données géométriques. Prenons pour exemple les mécanismes de perforation ou de libération des grains de sécrétion mentionnés dans le modèle du flux continu de membranes (voir sections 5.2.3 et 6.5). Remarquons toutefois que la modélisation précise de la dynamique des compartiments n'est pas l'objet principal des langages biochimiques, plus centrés sur l'analyse formelle des propriétés émergentes des modèles (par exemple *via* le model checking).

L-systèmes et MGS

Dans [Terraz 09], les auteurs proposent une extension des L- systèmes basée sur les cartes généralisées de dimension 3. Ces travaux se place dans le contexte de la modélisation de la structure interne d'objets volumiques. Ils sont appliqués à la construction de plantes, plus exactement d'arbres. Le langage proposé s'appuie notamment sur des contraintes topologiques fortes propres aux arbres. En particulier, les volumes manipulés sont tous des prismes.

Notons que les auteurs signalent qu'ils se placent dans un contexte comparable à MGS [Spicher 06], un langage de programmation dédié à la modélisation et à la simulation de systèmes biologiques. Ils précisent alors que MGS propose parmi ses structures de données, des structures topologiques plus générales que les arbres. Le langage MGS propose des primitives de programmation exploitant la recherche et la modification de motifs basées sur le parcours de structures topologiques. En particulier, le langage MGS s'appuie sur l'outil Moka pour implémenter les cartes généralisées. A notre connaissance, le langage MGS ne permet pas la description de motifs génériques, comme le permettent nos règles de transformation avec variables. Une piste peut-être intéressante pourrait être de programmer nos règles de transformation en langage MGS.

7.2 Vers un simulateur biologique à base topologique

Dans cette section, nous présentons le cahier des charges de notre simulateur biologique à base topologique. Pour cela, nous nous appuyons sur le développement de notre premier prototype de simulateur basé sur un noyau de 3-G-cartes. Comme nous l'avons vu dans la première partie de ce manuscrit, le choix des cartes généralisées nous assure de bien prendre en compte la structuration de nos objets. Dans le cadre de la biologie, les modèles topologiques nous permettent de nous focaliser sur les relations de voisinages qui existent entre les différents compartiments. De plus, ils présentent l'avantage de savoir capturer la structure des objets avec peu d'informations géométriques. Par exemple, dans la section 6.5.2 page 159, nous avons vu que la tour ainsi que le pile d'assiettes – en dépit de leur forme simple – mettent en lumière les distinctions topologiques qui existent entre les modèles de l'appareil de Golgi.

Afin d'illustrer au mieux le cahier des charges du simulateur, nous présentons en section 7.2.1 deux modèles biologiques simples sur lesquels nous nous appuyons dans le cahier des charges. Ces deux modèles font écho aux mécanismes mis en jeu dans l'appareil de Golgi.

Le cahier des charges de notre simulateur est découpé en plusieurs sections. En section 7.2.2, nous présentons le noyau de 3-G-cartes à la base du simulateur. Il comporte une structure de données pour les brins d'une 3-G-carte ainsi que quelques opérations élémentaires.

Dans la section 7.2.3, nous évoquons les différents plongements de nos objets topologiques. Dans notre cas, les données de plongement sont principalement de deux natures : les données biochimiques modélisent les molécules tandis que les données géométriques – même basiques – nous permettent de visualiser les objets dans l'espace. Dans cette même section, nous étudions les différentes sorties graphiques que nous mettons à disposition des utilisateurs de notre simulateur.

Dans la section 7.2.4 nous proposons une structure de données pour les règles de transformation de 3-G-cartes. Dans cette même section, nous étudions également les mécanismes d'application des règles.

7.2.1 Deux modèles biologiques

Transport de protéines le long d'un tube

Dans ce premier modèle, que nous dénommons *modèle du transport*, nous nous intéressons au transport de protéines le long d'un tube. Ce mécanisme intervient notamment dans le modèle du flux continu de membranes (voir section 5.2.3), où des tubes connectent les saccules. Nous considérons deux espèces de protéines : une espèce de protéines solubles qui se déplace à l'intérieur des compartiments et une espèce de protéines membranaires qui se déplacent sur leur membrane. Une représentation en deux dimensions de ce phénomène biologique est proposée en Fig. 7.5.

Sur la Figure 7.5, on remarque deux compartiments A et B connectés par un tube. Nous considérons que le tube est un compartiment au même titre que A et B. Remarquons que les traits en pointillés qui

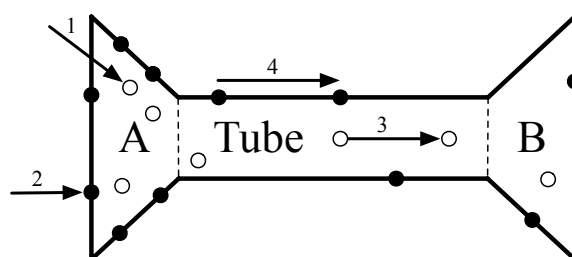


FIGURE 7.5 – Modèle du transport

permettent de distinguer les compartiments sur la figure n'ont pas d'existence biologique. On suppose que dans la réalité, aucune membrane ne sépare A et B du tube. Les protéines solubles – qui évoluent à l'intérieur des compartiments – sont représentées par de petits cercles. Les protéines membranaires, représentées par de petits points noirs, sont enchâssées sur la membrane des compartiments.

Quatre règles de comportement capturent la dynamique de ce mécanisme biologique. Sur la figure, les règles sont représentées par des flèches numérotées :

1. apparition de protéines solubles dans le compartiment A. Notons que dans notre modèle, la provenance des protéines – membranaires comme solubles – importe peu ;
2. apparition de protéines membranaires à la surface du compartiment A ;
3. transport des protéines solubles. Elles circulent au sein des compartiments et peuvent également passer d'un compartiment à un autre, par exemple de A au tube. On suppose que le transport ne suit pas une direction privilégiée ;
4. transport des protéines membranaires selon le même principe.

Bourgeonnement des vésicules

Notre second modèle, que nous dénommons *modèle du bourgeonnement*, est également inspiré de l'appareil de Golgi. Ici, nous nous intéressons au bourgeonnement des vésicules à la surface des saccules. Ce mécanisme complexe intervient dans le modèle vésiculaire (voir section 5.2.1) ainsi que dans le modèle de la maturation (voir section 5.2.2). Nous l'étudions ici dans une version très simplifiée. Une représentation en deux dimensions de ce mécanisme est proposée en Fig. 7.6.

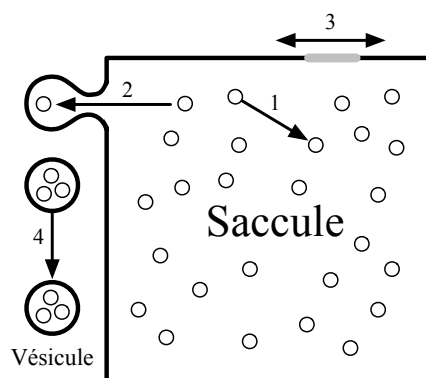


FIGURE 7.6 – Modèle du bourgeonnement

Nous considérons deux types de compartiments : un saccule – vu de dessus sur la figure – et des vésicules situées à sa périphérie. Des protéines solubles, d’une seule espèce, diffusent à l’intérieur du saccule. Tandis que nous considérons toujours un unique saccule, le nombre de vésicules augmente au fur et à mesure des bourgeonnements. Enfin, enchâssée sur la membrane du saccule, on remarque une zone plus épaisse colorée en gris ; il s’agit d’un activateur responsable du bourgeonnement des vésicules (voir section 5.2.1). Dans notre modèle, nous considérons un unique activateur, celui-ci se déplace le long de la membrane du saccule.

Quatre règles de comportement font évoluer la structure :

1. transport des protéines solubles. Comme dans le modèle précédent, on ne présuppose pas l’existence d’une direction de transport privilégiée ;
2. bourgeonnement d’une **Vésicule** à la surface du **Saccule**. Remarquons que des protéines du saccule sont évacuées par les vésicules ;
3. déplacement de l’activateur ;
4. déplacement des vésicules. Une fois détachées d’un saccule, les vésicules circulent dans le cytoplasme de la cellule.

Contrairement au modèle précédent, aucune règle ne renouvelle le stock de protéines dans la structure. Ainsi, après plusieurs bourgeonnements successifs, la quantité de protéines dans le saccule peut diminuer sensiblement.

7.2.2 Le noyau de 3-G-cartes

7.2.2.1 Notre structure de brins

Dans cette section, nous proposons une structure de données qui plante les brins, c’est-à-dire les éléments de base d’une G-carte (voir section 1.3.1). Dans la première partie du manuscrit, nous avons vu que les brins d’une 3-G-carte sont des sommets de graphe munis de quatre arcs incidents étiquetés respectivement par α_0 , α_1 , α_2 et α_3 . De plus, nous avons vu que des données de plongement peuvent être associées aux brins.

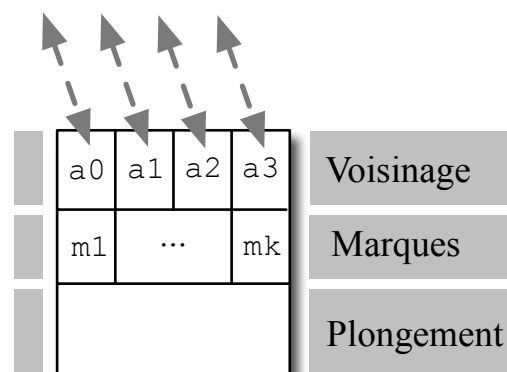


FIGURE 7.7 – Notre structure de brins

Notre structure de données est illustrée en Fig. 7.7. Afin de faciliter sa lecture, nous l’avons divisée en trois régions :

- dans la région « Voisinage », quatre pointeurs de brins a_0 , a_1 , a_2 et a_3 implantent respectivement les arcs α_0 , α_1 , α_2 et α_3 incidents aux brins ;

- la région « Marques » contient k marques entières $m1$ à m_k . Celles-ci sont principalement utilisées lors du parcours des brins d'une 3-G-carte ;
- la région « Plongement » contient les données relatives au plongement des brins. Comme elles varient en fonction de l'application, leur type n'est pas fixé *a priori*. Dans cette section, nous ignorons le plongement.

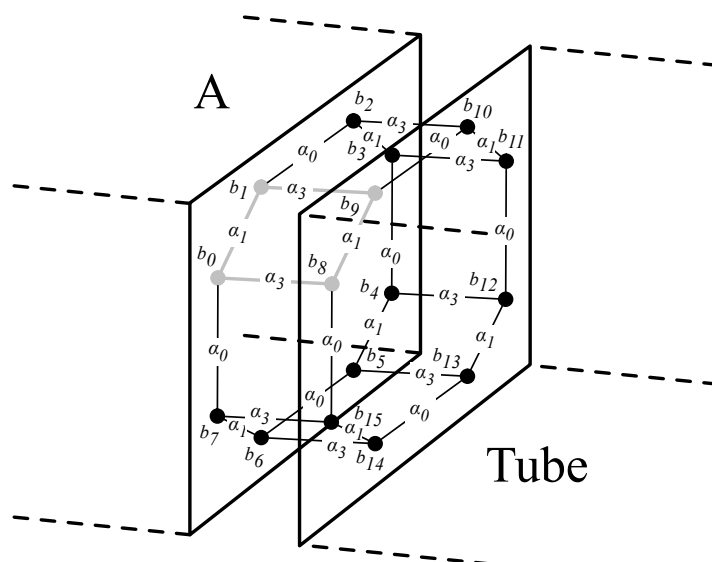


FIGURE 7.8 – Illustration de la 2-cellule qui connecte A au Tube

Illustrons à présent l'utilisation de notre structure de données dans le cadre du modèle du transport. Nous choisissons de représenter les compartiments A et B ainsi que le tube par des volumes. La Fig. 7.8 illustre la face – c'est-à-dire la 2-cellule – le long de laquelle sont connectés le compartiment A et le tube. Les brins b_0 à b_7 appartiennent à A tandis que les brins b_8 à b_{15} appartiennent au Tube. Rappelons que dans une 3-G-carte, deux volumes – c'est-à-dire deux 3-cellules – sont connectés par des liaisons α_3 . Ainsi, sur la figure, les brins b_0 à b_7 sont connectés deux à deux aux brins b_8 à b_{15} par α_3 .

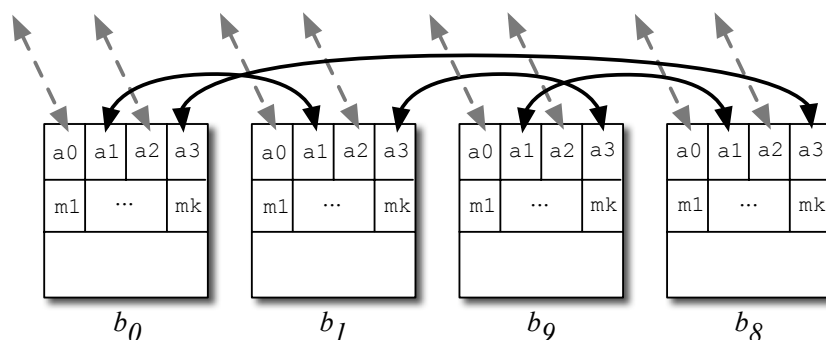


FIGURE 7.9 – Implantation de la partie grisée de la 3-G-carte de la Fig. 7.8

La Fig. 7.9 illustre l'implantation de la partie grisée de la Fig. 7.8. Elle comprend les brins b_0 , b_1 , b_8 et b_9 munis des arcs qui les connectent. Sur la Fig. 7.9, on remarque que le brin b_0 pointe sur b_1 via son pointeur $a1$. De la même manière, b_1 pointe sur b_0 via son pointeur $a1$. En effet, rappelons que dans une

G -carte, les liens qui existent entre deux brins sont toujours bidirectionnels. Sur la figure, les brins b_1 et b_9 sont connectés deux à deux *via* leur pointeur a_3 . Il en est de même pour les brins b_9 et b_8 , connectés par a_1 , et enfin pour les brins b_8 et b_1 , connectés par a_3 .

On remarque qu'en suivant successivement les pointeurs a_1 , a_3 , a_1 et a_3 à partir du brin b_0 , on revient sur b_0 . Il existe donc un cycle $\alpha_0\alpha_1\alpha_0\alpha_1$ qui passe par les brins b_0 , b_1 , b_8 , et b_9 . Cette propriété fait écho à la contrainte *Cycle* introduite à la définition 5 de la section 1.3.3. Dans le chapitre 4, nous avons vu que des critères syntaxiques permettent de nous assurer que les G -cartes résultats de l'application d'une règle de transformation préservent cette contrainte.

Nous venons d'introduire une structure de données qui implante les brins d'une 3- G -carte. Remarquons qu'un simple pointeur sur un brin n'est pas une structure de données acceptable pour les 3- G -cartes. Intuitivement, un unique brin ne permet pas d'atteindre tous les brins d'une 3- G -carte car celle-ci peut être composée de plusieurs composantes connexes. Nous pouvons par exemple planter une 3- G -carte par une liste de pointeurs sur un brin de chaque composante connexe. Nous pouvons également planter une 3- G -carte par la liste de tous ses brins, c'est le choix que nous avons fait dans notre premier prototype de simulateur.

7.2.2.2 Les opérations élémentaires du noyau

Comme chaque fois que l'on définit une nouvelle structure de données, il faut également définir les opérations de manipulation associées. Dans le cas de notre structure de brins, nous proposons les opérations suivantes : l'*ajout d'un brin*, la *suppression d'un brin*, la *connexion de deux brins*, la *déconnexion d'un brin*, des opérations de *parcours d'orbites* et des opérations de *marquage d'orbites*. Nous qualifions ces opérations d'*opérations élémentaires* en ce sens qu'elles n'assurent pas toutes la préservation des contraintes de cohérence des cartes généralisées.

Ajout et suppression d'un brin

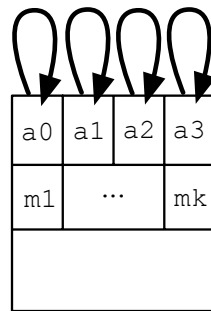


FIGURE 7.10 – Implantation d'un brin isolé

L'opération d'ajout d'un brin implante l'opération de même nom introduite en section 1.4.1. Il s'agit d'ajouter un brin isolé – c'est-à-dire un brin qui possède quatre arcs incidents étiquetés par α_0 , α_1 , α_2 et α_3 – à une carte généralisée. La Fig. 7.10 illustre l'implantation un brin isolé. Lorsque l'on ajoute un nouveau brin, celui-ci n'est pas marqué. Sur la figure, les marques sont toutes à 0. Enfin, remarquons que l'opération d'ajout d'un brin préserve les contraintes de cohérence des 3- G -cartes.

L'opération de suppression d'un brin implante l'opération de même nom introduite en section 1.4.2. Ici, on retire un brin isolé dans une carte généralisée. Comme l'opération d'ajout d'un brin, celle-ci préserve les contraintes de cohérences des cartes généralisées.

Connexion et déconnexion de deux brins

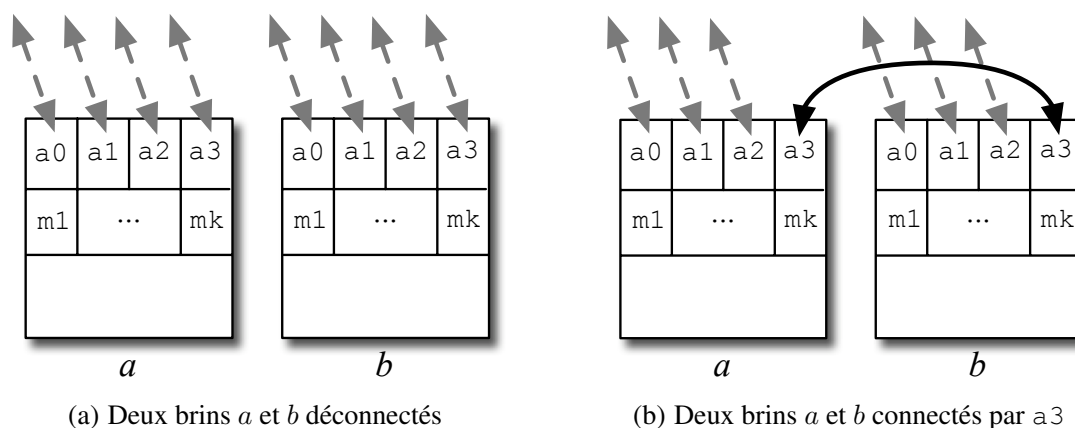


FIGURE 7.11 – Illustration de l'opération de connexion de deux brins

L'opération de connexion permet de connecter deux brins par a_i (avec $0 \leq i \leq 3$). Elle est illustrée en Fig. 7.11. Sur la Fig. 7.11(a) sont représentés deux brins *a* et *b* qui ne sont pas connectés, c'est-à-dire qui ne pointent pas l'un sur l'autre. Le résultat de l'application de l'opération de connexion de *a* et *b* par a_3 est illustré en Fig. 7.11(b). Après la connexion, *a* pointe sur *b* par a_3 et *b* pointe sur *a* par a_3 . On remarque que cette opération élémentaire nous assure que les liens qui existent entre les brins sont bidirectionnels. L'opération de connexion de deux brins ne préserve pas la contrainte *Cycle* des 3-G-cartes. En particulier, elle se distingue de l'opération de couture (voir section 1.4.3).

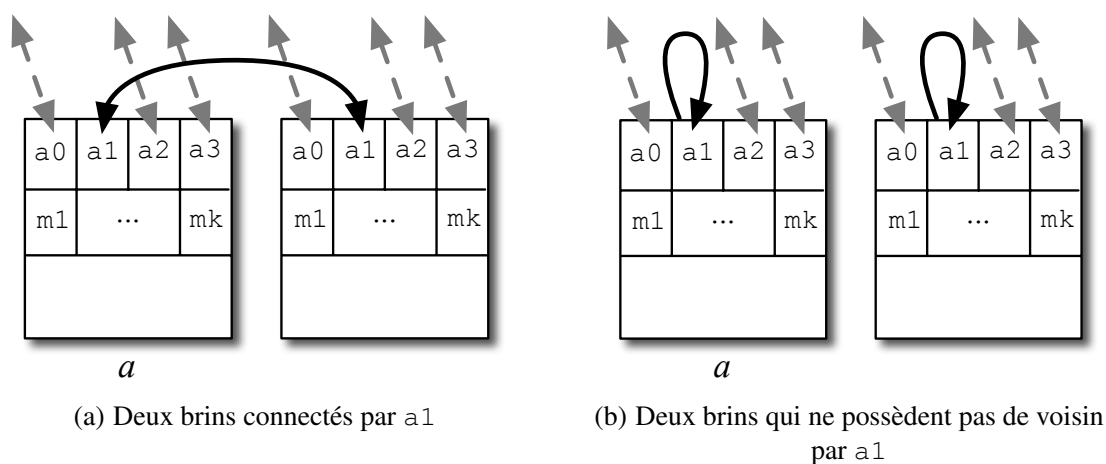


FIGURE 7.12 – Illustration de l'opération de déconnexion d'un brin

L'opération de déconnexion d'un brin par a_i (avec $0 \leq i \leq n$) déconnecte un brin de son brin voisin par a_i . Illustrons cette opération à l'aide de la Fig. 7.12. En déconnectant le brin *a* de la Fig. 7.12(a) par a_1 , on obtient la configuration illustrée en Fig. 7.12(b). Remarquons que si le brin *a* est initialement connecté à lui-même par a_1 , aucune modification n'est effectuée.

Comme l'opération précédente, la déconnexion d'un brin ne préserve pas la contrainte *Cycle* des 3-G-cartes.

Parcours et marquage

Dans les modeleurs géométriques à base topologique, la notion de *parcours* joue un rôle important. L'un des avantages des modèles topologiques est qu'ils représentent explicitement le voisinage des objets. Les opérations de parcours d'orbites permettent d'appliquer des traitements sur les brins d'une orbite. Par exemple, dans le cas d'une 3-G-carte munie d'un plongement polyédrique (voir section 1.2), le parcours des brins d'une face permet de récupérer les sommets géométriques de la face. Nous distinguons deux types de parcours : le parcours générique et les parcours pré-construits.

```

1 let parcours_gén brin liste_alpha traitement =
2   (* création de la liste des brins marqués lors du parcours *)
3   let brins_marqués = liste_brins () in
4
5   (* fonction auxiliaire récursive *)
6   let rec parcours_gén_aux brin =
7     if (non_marqué brin) then (
8       (* traitement du brin courant *)
9       marquer brin ;
10      ajouter_brin brins_marqués brin ;
11      traitement brin ;
12
13      (* appel récursif sur les voisins du brin *)
14      List.iter
15        (fun
16          0 -> parcours_gén_aux !^(brin.a0)
17          | 1 -> parcours_gén_aux !^(brin.a1)
18          | 2 -> parcours_gén_aux !^(brin.a2)
19          | 3 -> parcours_gén_aux !^(brin.a3)
20        )
21      liste_alpha ;
22    done ;
23  ) in
24
25  (* appel de la fonction auxiliaire récursive *)
26  parcours_gén_aux brin
27
28  (* démarquage des brins *)
29  List.iter (fun brin -> démarquer brin) brins_marqués

```

Code 7.1 – Parcours générique

L'algorithme de parcours générique des brins d'une orbite est illustré dans le Code 7.1. La fonction `parcours_gén` prend trois paramètres : le brin de départ du parcours, une liste `liste_alpha` qui modélise l'orbite à parcourir³ et une fonction de `traitement` à appliquer sur les brins parcourus. Intuitivement, la fonction `parcours_gén` implante un parcours en profondeur. La terminaison du parcours est assurée par un marquage des brins traités (voir les lignes 7 à 10). La fonction auxiliaire `parcours_gén_aux` est appelée récursivement sur les brins voisins du brin courant *via* une itération sur la liste `liste_alpha` (voir les lignes 14 à 21).

3. Par exemple, nous représentons l'orbite $\langle \alpha_0 \alpha_1 \alpha_2 \alpha_3 \rangle$ – qui permet de parcourir une composante connexe dans une 3-G-carte – par la liste d'entiers `[0 ; 1 ; 2 ; 3]`.

```

1 let parcours_01 brin traitement =
2   (* création de la liste des brins marqués lors du parcours *)
3   let brins_marqués = liste_brins () in
4
5   (* fonction auxiliaire récursive *)
6   let parcours_01_aux brin =
7     if (non_marqué brin) then (
8       (* traitement du brin courant *)
9       marquer brin ;
10      ajouter_brin brins_marqués brin ;
11      traitement brin ;
12
13      (* traitement du voisin par alpha_0 *)
14      let voisin_0 = !^(brin.a0) in
15      marquer voisin_0 ;
16      ajouter_brin brins_marqués voisin_0 ;
17      traitement voisin_0 ;
18
19      (* appel récursif sur le voisin par a1 de voisin_0 *)
20      parcours_01_aux !^(voisin_0.a1) ;
21    ) in
22
23   (* appel de la fonction auxiliaire récursive *)
24   parcours_01_aux brin ;
25
26   (* démarquage des brins *)
27   List.iter (fun brin -> démarquer brin) brins_marqués

```

Code 7.2 – Parcours d’une orbite $\langle \alpha_0 \alpha_1 \rangle$

Dans les 3-G-cartes, certains parcours sont souvent utilisés. C’est le cas du parcours de l’orbite $\langle \alpha_0 \alpha_1 \rangle$, utilisée pour récupérer les sommets géométriques d’une face⁴. Dans ce cas, il convient d’implanter des algorithmes spécifiques plus efficaces qu’un parcours en profondeur générique. Le parcours de l’orbite $\langle \alpha_0 \alpha_1 \rangle$ est illustré par le Code 7.2. La fonction `parcours_01` prend deux paramètres : le `brin` de départ du parcours et une fonction de `traitement` à appliquer sur les brins parcourus. Intuitivement, il s’agit de parcourir les brins en séquence : on traite le brin de départ (ligne 11), puis son voisin par α_0 (ligne 17) puis on appelle la fonction `parcours_01_aux` récursivement sur le voisin par $\alpha_0 \alpha_1$ (ligne 20). À nouveau, la terminaison est assurée par un marquage des brins traités (lignes 7 à 10, 15 et 16). Remarquons que le parcours de l’orbite $\langle \alpha_0 \alpha_1 \rangle$ peut également être mis en oeuvre par la fonction `parcours_gén` via l’appel suivant : `parcours_gén brin [0;1] traitement`. Cependant, ce parcours est moins efficace que le parcours spécifique implanté dans la fonction `parcours_01`. En moyenne, le parcours générique passe plus souvent sur des brins déjà marqués.

Enfin, remarquons que lors d’un parcours, la conservation des contraintes de cohérence est tributaire des modifications locales effectuées par la fonction `traitement`. Par exemple, si la fonction `traitement` modifie les connexions des brins, la contrainte *Cycle* peut ne plus être satisfaite.

De la même manière que les parcours, les opérations de marquage sont classiques en modélisation géométrique à base topologique. Dans les 3-G-cartes, nous les implantons par les opérations de par-

4. Comme les volumes sont collés le long d’une face commune par des liaisons α_3 , deux brins liés par α_3 possèdent le même plongement géométrique. Il est donc inutile d’emprunter les liaisons α_3 lors du parcours.

cours. Intuitivement, la fonction `traitement` passée en paramètre des parcours permet de modifier les marques des brins. Remarquons ici l'importance d'associer un tableau de marques à chaque brin. En effet, comme nous venons de le voir, les algorithmes de marquage utilisent déjà une marque.

7.2.3 Plongements et sorties graphiques

Le plongement d'une 3-G-carte permet d'associer des données aux orbites. Dans notre cas, outre des données géométriques – qui permettent notamment d'implanter des primitives d'affichage des objets – nous souhaitons manipuler des données biochimiques telles que des quantités de protéines. Dans cette section, à partir des modèles biologiques introduits dans la section 7.2.1, nous montrons de quelle manière sont construites les structures de données qui contiennent les plongements. Ensuite, nous évoquons les différentes sorties graphiques que nous proposons à l'utilisateur du simulateur.

7.2.3.1 Plongement

Dans cette section, nous détaillons principalement le plongement des volumes. Trois autres orbites sont plongées dans notre simulateur : les demi-faces (c'est-à-dire les orbites $\langle \alpha_0 \alpha_1 \rangle$), les arêtes simples (c'est-à-dire les orbites $\langle \alpha_0 \rangle$) et les 0-cellules. Les plongements de ces trois cellules sont respectivement détaillés en annexes B.1, B.2 et B.3. Notons que les plongements des demi-faces et des arêtes simples sont analogues aux plongements des volumes : ils contiennent principalement des données biochimiques. En revanche, le plongement des 0-cellules contient uniquement des données géométriques. Notons enfin que l'annexe B.4 contient des exemples de plongements utiles pour la modélisation du bourgeonnement des vésicules.

Dans le modèle du transport, nous manipulons trois compartiments : deux compartiments A et B, et un tube (voir Fig. 7.5 page 173). Chacun de ces compartiments contient des protéines solubles. Dans le simulateur, nous modélisons les compartiments biologiques par des volumes, c'est-à-dire par des 3-cellules. Le modèle du transport nous mène à associer deux données de plongement à chaque compartiment : un *type biologique* – ou plus simplement *type* – qui nous permet de distinguer les trois compartiments et une quantité de protéines solubles. Le typage des volumes est fondamental, il nous permet d'écrire des règles de transformation dédiées à certains compartiments. Par exemple, dans le modèle du transport, nous souhaitons écrire une règle qui synthétise des protéines membranaires dans le compartiment A. Remarquons que le typage dépend du modèle considéré. Par exemple, les types qui caractérisent les compartiments A, B et le tube n'ont pas de sens dans le modèle du bourgeonnement. En terme d'implantation, ceci signifie que le plongement des volumes est en partie déterminé par l'utilisateur.

```
1 type t_plongement_volume =
2   Volume_Défaut
3   | Volume of t_plongement_volume_utilisateur
```

Code 7.3 – Type Caml pour le plongement des volumes

Le plongement des volumes est une valeur du type Caml⁵ `t_plongement_volume`, nous l'introduisons dans le Code 7.3. Le type `t_plongement_volume` présente deux constructeurs. Le constructeur constant `Volume_Défaut` est le type biologique que l'on utilise lorsque l'utilisateur ne précise pas explicitement de type biologique. Le constructeur `Volume` prend en paramètre une valeur du type Caml `t_plongement_volume_utilisateur`, celui-ci dépend du modèle.

5. Le terme *type Caml* est à distinguer du terme *type biologique*, il désigne le type des valeurs Caml.

```

1 type t_plongement_volume_utilisateur =
2   Volume_A of (t_protéine list)
3   | Volume_B of (t_protéine list)
4   | Volume_Tube of (t_protéine list)

```

Code 7.4 – Type Caml utilisateur pour le plongement des volumes dans le modèle du transport

Le type Caml `t_plongement_volume_utilisateur` utilisé dans le modèle du transport est introduit dans le Code 7.4, il est écrit par l'utilisateur et implante les types biologiques du modèle. Intuitivement, à chaque type biologique correspond un constructeur. Ainsi, les types biologiques spécifiques au modèle du transport sont : `Volume_A`, `Volume_B` et `Volume_Tube`. On remarque que chacun de ces constructeurs prend une liste de `t_protéine` en paramètre. C'est de cette manière que les protéines solubles sont associées aux compartiments. Dans la pratique, chaque élément d'une liste de protéines correspond à la quantité – ou bien à la concentration – d'une espèce protéique dans le compartiment considéré⁶. Rappelons que dans l'exemple du transport de protéines le long d'un tube, nous considérons une seule espèce de protéines solubles. Les listes passées en paramètre des constructeurs `Volume_A`, `Volume_B` et `Volume_Tube` contiennent donc un seul élément. Nous reviendrons sur le type Caml `t_protéine` dans la dernière section de ce chapitre. Remarquons enfin qu'un volume de type biologique `Volume_Défaut` ne contient pas de protéines.

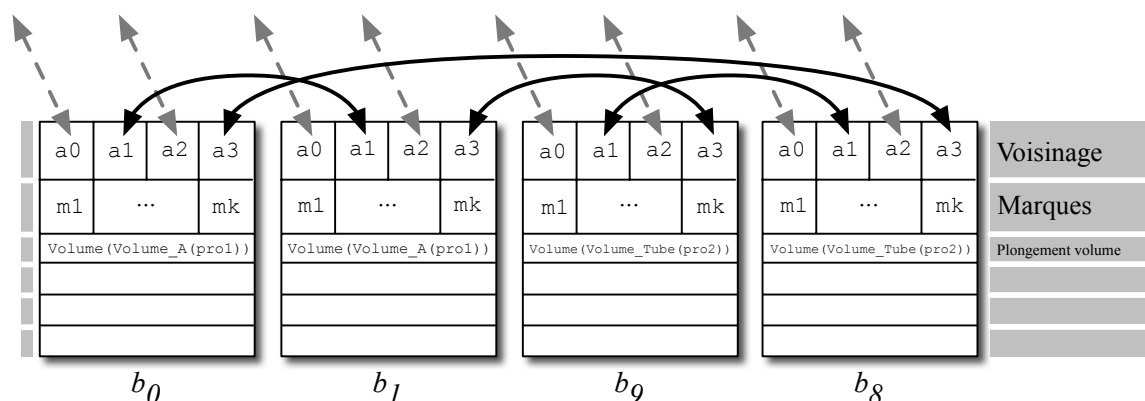


FIGURE 7.13 – Exemple de plongement des volumes

Sur la Fig. 7.13, nous reprenons les brins b_0 , b_1 , b_8 et b_9 de la Fig. 7.8 page 175 et leur attribuons une valeur de plongement volume. Les brins b_0 et b_1 appartiennent au volume qui modélise le compartiment A, il présente donc le même plongement de volume : `Volume (Volume_A (pro1))`. De même, les brins b_8 et b_9 – qui appartiennent au compartiment tube – portent tous les deux le plongement de volume `Volume (Volume_Tube (pro2))`. Dans les deux cas, `pro1` et `pro2` sont des valeurs du type Caml `t_protéine list`. Remarquons que comme l'impose la contrainte *Plongement* introduite dans le chapitre 4, deux brins qui appartiennent au même volume possèdent le même plongement de volume. Comme les contraintes de cohérence des cartes généralisées – c'est-à-dire les contraintes *Incide* et *Cycle* – la contrainte *Plongement* est assurée par les règles de transformation.

Sur la Fig. 7.13, on remarque que les brins portent tous une donnée de plongement de volume. L'avantage de cette implantation est que le plongement d'un volume donné peut-être consulté depuis tous les brins du volume. D'autres implantations sont possibles. Par exemple, nous pouvons faire porter le plongement de volume par un unique brin de chaque volume. Cette implantation, rendue possible par

6. Dans la suite du chapitre, nous manipulons des quantités de protéines plutôt que des concentrations.

a0	a1	a2	a3	Voisinage
m1	...		mk	Marques
plongement_volume				Plongement
plongement_demi_face				
plongement_arête_simple				
plongement_sommet				

FIGURE 7.14 – Notre structure de brins munie de ses données de plongement

la contrainte *Plongement*, présente l'avantage de minimiser l'espace mémoire utilisé par les données de plongement. Cependant, dans ce cas, un parcours de volume est nécessaire pour retrouver le brin qui porte le plongement. Une implantation du plongement par des pointeurs permet de concilier les deux méthodes. Dans ce cas, à chaque brin est associé un pointeur sur sa donnée de plongement de volume. De cette manière, la donnée de plongement n'est pas dupliquée et reste consultable depuis tous les brins. Remarquons que cette méthode est d'autant plus avantageuse que l'espace mémoire requis par la donnée de plongement est grande devant la taille mémoire d'un pointeur.

La structure complète d'un brin plongé est enfin illustrée en fig. 7.14. À présent, quatre champs figurent dans la région « Plongement » et permettent d'accéder aux données de plongement d'un brins.

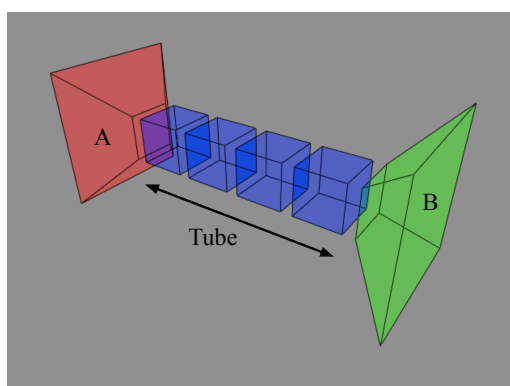
7.2.3.2 Les sorties graphiques

Afin de faciliter la prise en main de notre simulateur, nous proposons plusieurs modes de visualisation des objets. Nous avons mis en oeuvre l'ensemble de ces visualisations dans notre prototype de simulateur. Une première façon de montrer les objets à l'utilisateur consiste à les afficher dans l'espace. L'affichage 3D des objets est rendu possible par les sommets géométriques qui plongent les 0-cellules. La Fig. 7.15 illustre ce premier mode de visualisation. La colonne de gauche (Fig. 7.15(a), 7.15(c), 7.15(e) et 7.15(g)) correspond au modèle du transport, la seconde (Fig. 7.15(b), 7.15(d), 7.15(f) et 7.15(h)) correspond au modèle du bourgeonnement.

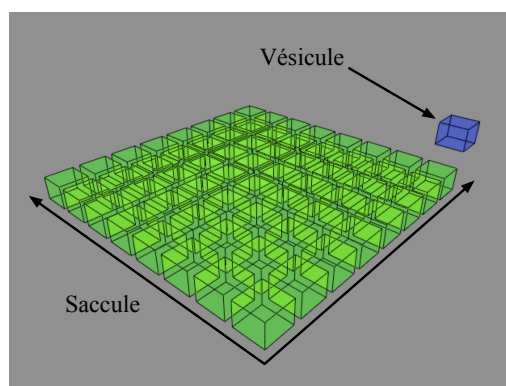
Nous proposons deux modes de visualisation des objets dans l'espace : la visualisation de l'intérieur des compartiments (c'est-à-dire les volumes) et la visualisation de leur membrane (c'est-à-dire les demi-faces). Les compartiments des deux modèles biologiques sont illustrés sur les Fig. 7.15(a) et 7.15(b). Nous avons implanté l'affichage en OpenGL⁷ [OGL]. Dans notre prototype, nous proposons d'associer une couleur à chaque type de compartiments. Par exemple, sur la Fig. 7.15(a), les compartiments A, B et le tube sont représentés avec des couleurs distinctes. De même, la vésicule de la Fig. 7.15(b) se distingue du saccule. On remarque que le tube ainsi que le saccule sont discrétisés en plusieurs petits volumes. Cette discrétisation nous permet d'associer des protéines localement à l'intérieur des compartiments. Ainsi, chaque volume de tube (respectivement de saccule) porte une donnée de plongement propre munie de ses propres protéines. Le transport des protéines d'une section de tube (respectivement de saccule) à une autre est permis par des demi-faces virtuelles. Celles-ci sont représentées en gris foncé sur les Fig. 7.15(c) et 7.15(d).

Les Fig. 7.15(e) et 7.15(f) illustrent le mode de visualisation des membranes. De même que pour les volumes, le simulateur propose d'associer différentes couleurs aux membranes en fonction de leur type. Sur la Fig. 7.15(e) se distinguent les membranes des compartiments A, B et du tube. Remarquons que

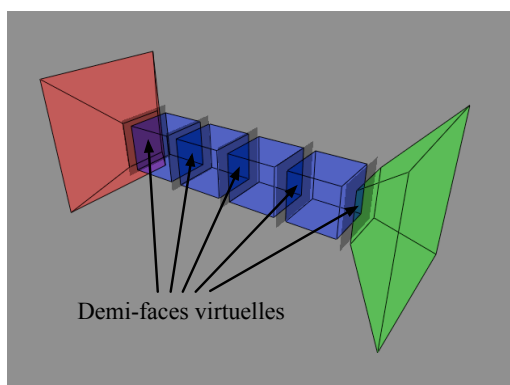
7. OpenGL est une bibliothèque qui contient des primitives pour l'affichage d'objets dans l'espace.



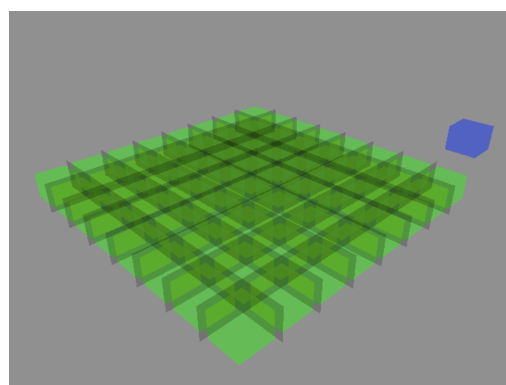
(a) Compartiments du premier modèle



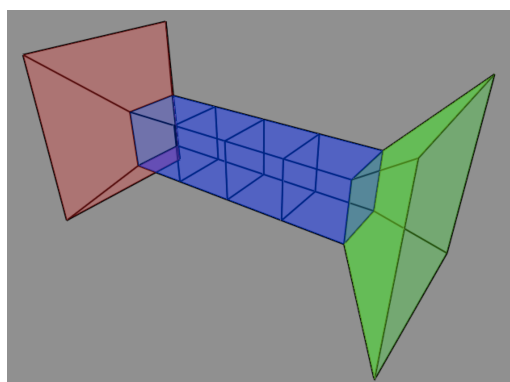
(b) Compartiments du second modèle



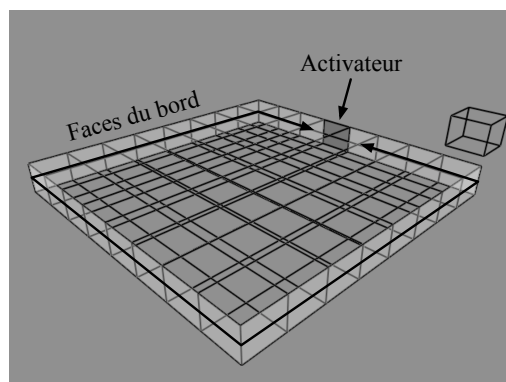
(c) Demi-faces virtuelles du premier modèle



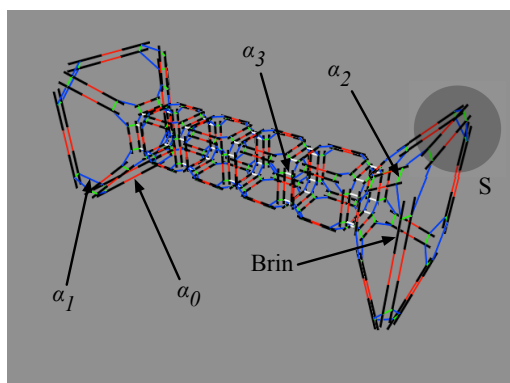
(d) Demi-faces virtuelles du second modèle



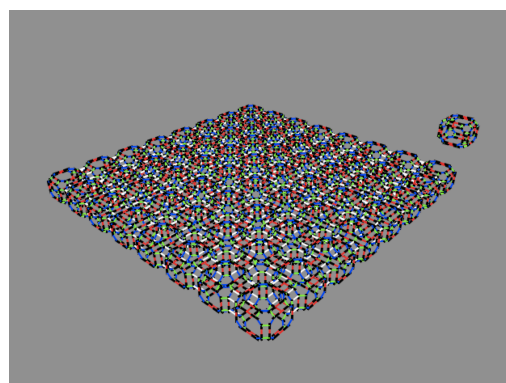
(e) Membranes du premier modèle



(f) Membranes du second modèle



(g) Vue topologique du premier modèle



(h) Vue topologique du second modèle

FIGURE 7.15 – Visualisation des objets dans les modèles du transport (premier modèle, colonne de gauche) et du bourgeonnement (second modèle, colonne de droite)

par défaut, les demi-faces virtuelles ne sont pas affichées dans le mode de visualisation des membranes. Sur la Fig. 7.15(f), les demi-faces de type `Demi_Face_Bord` apparaissent en gris clair. La demi-face de type `Demi_Face_Bord_Activateur` est colorée en gris foncé. Les autres demi-faces, de type `Demi_Face_Virtuelle` ou `Demi_Face_Défaut`, ne sont pas colorées.

Les Fig. 7.15(g) et 7.15(h) illustrent le dernier mode de visualisation du simulateur : le mode de visualisation de la topologie. Ici, la structure topologique du modèle biologique étudié – c'est-à-dire sa 3-G-carte – est représentée explicitement. Les brins sont représentés par de petits segments épais et les arcs $\alpha_0, \alpha_1, \alpha_2$ et α_3 sont représentés par des segments colorés qui connectent les brins. On remarque sur la Fig. 7.15(h) que la quantité d'éléments à afficher rend la visualisation difficile à interpréter. Cependant, cette visualisation de la topologie présente l'avantage de permettre le contrôle du résultat des transformations topologiques.

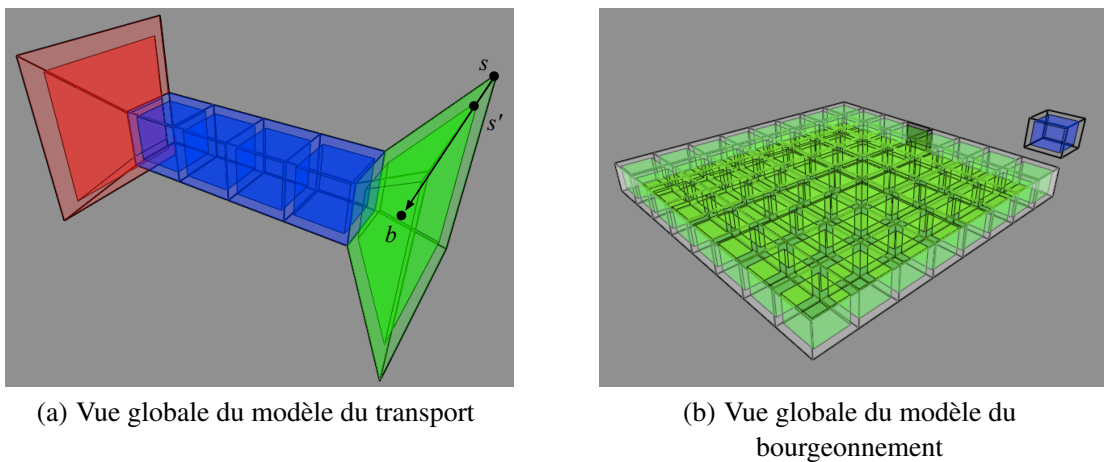


FIGURE 7.16 – Visualisation simultanée des compartiments et des membranes

Enfin, remarquons que le simulateur permet également d'afficher de concert les compartiments ainsi que leur membrane. Ce mode de visualisation est illustré dans les Fig. 7.16(a) et 7.16(b). Sur les figures, on remarque que les compartiments apparaissent à l'intérieur de leur membrane. Intéressons-nous aux sommets s' et s qui, sur la Fig. 7.16(a), correspondent respectivement à un sommet du compartiment B et à un sommet de sa membrane. Dans la structure topologique de la Fig. 7.15(g), ces deux sommets sont associés à une seule 0-cellule notée S . Nous savons qu'un unique sommet géométrique est associé à chaque 0-cellule par l'intermédiaire du plongement. Dans la pratique, ce sommet géométrique est s et le sommet s' est obtenu en effectuant un décalage de s . Le décalage est calculé par les primitives d'affichage que nous avons développé dans le prototype de simulateur. Intuitivement, les sommets des compartiments sont obtenus en décalant les sommets des membranes à l'aide des barycentres des volumes. Ce mécanisme est illustré par la flèche grise de la Fig. 7.16(a) : s' est calculé en rapprochant s du barycentre b du compartiment B.

La Fig. 7.17 illustre l'intérêt de la discrétisation des compartiments lorsque l'on modélise le transport de protéines dans un compartiment biologique. Les Fig. 7.17(a), 7.17(b) et 7.17(c) montrent un modèle simple d'appareil de Golgi connecté, elles sont issues d'une simulation où l'on transporte des protéines solubles depuis la région *cis* (en base sur les figures) vers la région *trans*. Sur la Fig. 7.17(a), on remarque des protéines qui entrent dans l'appareil de Golgi par la région *cis*. Sur la Fig. 7.17(b), les protéines commencent à migrer le long des tubes qui connectent les saccules. Enfin, sur la Fig. 7.17(c), les protéines ont atteint le saccule médian et migrent vers la région *trans*. Le simulateur permet également de suivre l'évolution des quantités de protéines à l'intérieur des compartiments. Pour cela, il suffit de tracer les courbes d'évolution des données de plongement. De telles courbes sont analogues à celles qui, dans le

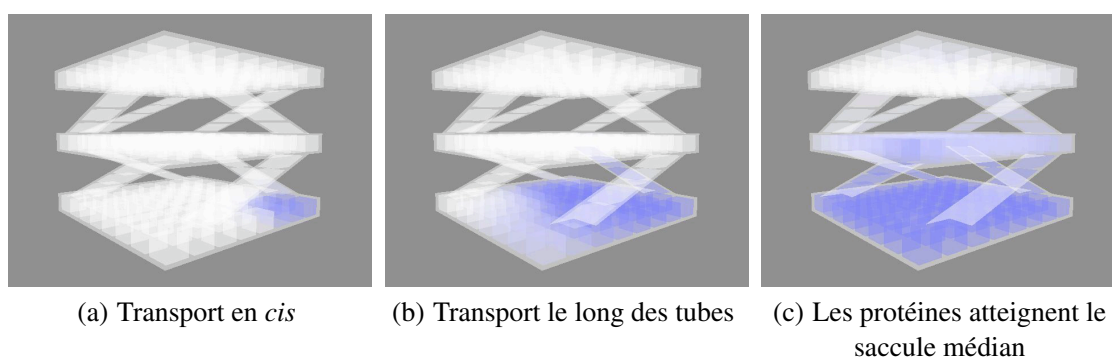


FIGURE 7.17 – Transport de protéines solubles dans un modèle de l'appareil de Golgi continu

chapitre 6, illustrent le résultat des simulations (voir par exemple la Fig. 6.8, page 151).

7.2.4 L'implantation des règles de transformation

7.2.4.1 Structure de données pour les règles de transformations

Dans notre premier prototype de simulateur, nous avons implanté les règles de transformation sans variable. Cette première implantation nous permet de poser les briques nécessaires au moteur d'application de règles du simulateur. Dans cette section, nous introduisons notre structure de données pour les règles de transformation de 3-G-cartes. En particulier, nous introduisons une structure pour les brins d'une règle. Cette dernière se distingue de la structure de brins de 3-G-cartes définie à la section 7.2.2.1. Dans la suite, lorsqu'il y a ambiguïté, nous employons les termes « brins de règles » et « brins de 3-G-cartes ».

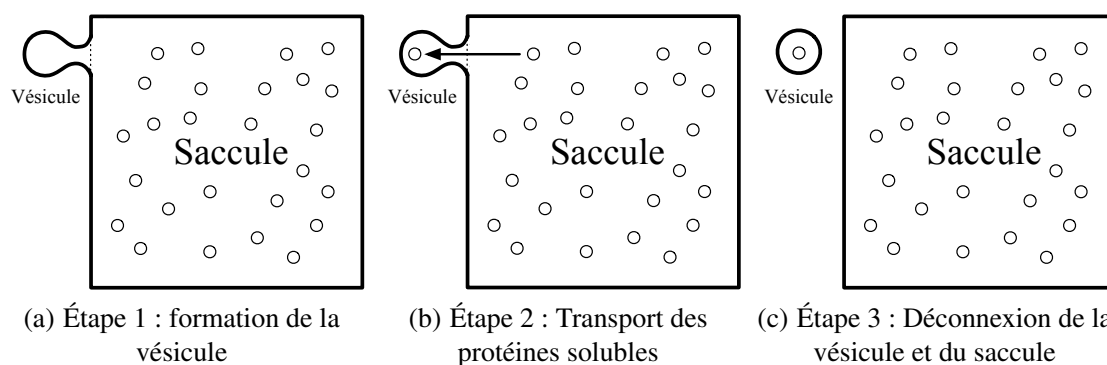


FIGURE 7.18 – Décomposition du mécanisme de bourgeonnement d'une vésicule

Afin d'introduire notre structure de brins de règles, considérons une règle de transformation issue du modèle du bourgeonnement. Parmi les quatre comportements introduits dans cet exemple (voir section 7.2.1), l'un d'eux est responsable du bourgeonnement. En Fig. 7.18, nous décomposons le bourgeonnement en trois étapes élémentaires :

- À l'étape 1 (voir Fig. 7.18(a)), le saccule est déformé et un nouveau compartiment – la vésicule – apparaît à sa surface. Remarquons que la vésicule reste connectée au saccule ; sur la figure, un trait en pointillés sépare la vésicule du saccule. Dans le simulateur, cette séparation est implantée par deux demi-faces virtuelles : la première appartient à la vésicule, la seconde appartient au saccule ;
- À l'étape 2 (voir Fig. 7.18(b)), quelques protéines solubles du saccule sont transportées dans la vésicule ;

- Lorsque la vésicule contient suffisamment de protéines, on entre dans l'étape 3 (voir Fig. 7.18(c)) et la vésicule se détache du saccule. En terme d'implantation, lors de l'étape 3, la demi-face virtuelle de la vésicule devient une demi-face de type `Demi_Face_Défaut` et la demi-face virtuelle du saccule devient une demi-face de type `Demi_Face_Bord`.

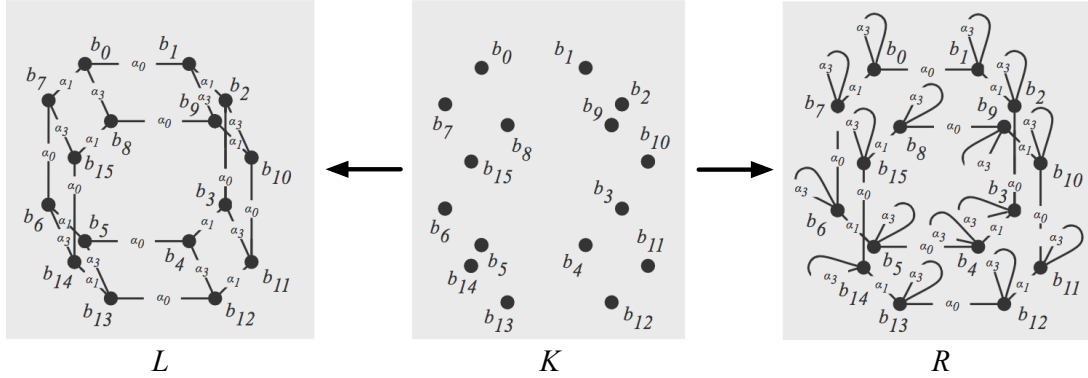


FIGURE 7.19 – La règle de déconnexion

La règle de transformation de la Fig. 7.19 modélise l'étape 3, nous l'appelons *règle de déconnexion*. Les brins b_0 à b_7 appartiennent à la vésicule tandis que les brins b_8 à b_{15} appartiennent au saccule. Dans la partie gauche L de la règle, les demi-faces $\langle \alpha_0 \alpha_1 \rangle (b_0)$ et $\langle \alpha_0 \alpha_1 \rangle (b_8)$ sont connectées. Elles apparaissent déconnectées dans la partie droite R . Pour plus de clarté, les plongements – c'est-à-dire les étiquettes des sommets – n'apparaissent pas sur la figure. Cependant, la règle modifie de concert la topologie et le plongement du modèle biologique. Par exemple, le type des deux demi-faces est modifié. Rappelons que dans notre simulateur, nous plongeons les orbites suivantes :

- les volumes $\langle O_v \rangle = \langle \alpha_0 \alpha_1 \alpha_2 \rangle$;
- les demi-faces $\langle O_f \rangle = \langle \alpha_0 \alpha_1 \rangle$;
- les arêtes-simples $\langle O_a \rangle = \langle \alpha_0 \rangle$;
- les sommets $\langle O_s \rangle = \langle \alpha_1 \alpha_2 \alpha_3 \rangle$.

Dans L , les plongements sont définis de la manière suivantes :

- $\Pi_L^{\langle O_v \rangle}(b_i) = \text{Volume}(\text{Volume_Vésicule}(p))$, $i \in [0, 7]$ et p une variable du type `Caml t_protéine list` ;
- $\Pi_L^{\langle O_v \rangle}(b_j) = v_j$, $j \in [8, 15]$ et v_j une variable du type `Caml t_plongement_volume` ;
- $\Pi_L^{\langle O_f \rangle}(b_k) = f_k$, $k \in [0, 15]$ et f_k une variable du type `Caml t_plongement_demi_face` ;
- $\Pi_L^{\langle O_a \rangle}(b_k) = a_k$, $k \in [0, 15]$ et a_k une variable du type `Caml t_plongement_arête_simple` ;
- $\Pi_L^{\langle O_s \rangle}(b_k) = s_k$, $k \in [0, 7, 8, 15]$ et s_k une variable du type `Caml t_plongement_sommet`.

Seul le plongement O_v pour les brins b_0 à b_7 est filtré explicitement et nous assure que le volume $\langle O_v \rangle (b_0)$ modélise une vésicule. Pour toutes les autres données de plongement, nous utilisons des variables qui – par définition – permettent de filtrer n'importe quelle donnée. Notamment, les plongements de L n'assurent pas explicitement que le volume $\langle O_v \rangle (b_8)$ modélise un saccule. Dans la pratique, nous filtrons la vésicule et supposons que par construction, une vésicule ne peut-être collée qu'à un saccule. De même, nous supposons que les demi-faces qui séparent une vésicule d'un saccule sont toujours des demi-faces virtuelles.

Dans le chapitre 4, nous avons vu que les variables de plongement de L sont instanciées lors du filtrage de la règle. De cette manière, lorsque l'on remplace L par R dans le graphe que l'on transforme,

les variables de L ont toutes une valeur. De fait, les variables de plongement de R font référence aux variables de plongement de L après le filtrage. Ainsi, nous définissons les plongements de R de la manière suivante :

- $\Pi_L^{O_v}(b_i) = \text{Volume}(\text{Volume_Vésicule}(p))$, $i \in [0, 7]$;
- $\Pi_L^{O_v}(b_j) = v_j$, $j \in [8, 15]$;
- $\Pi_L^{O_f}(b_i) = \text{Demi_Face_Défaut}$, $i \in [0, 7]$;
- $\Pi_L^{O_f}(b_j) = \text{Demi_Face}(\text{Demi_Face_Bord})$, $j \in [8, 15]$;
- $\Pi_L^{<O_a>}(b_k) = a_k$, $k \in [0, 15]$;
- $\Pi_L^{<O_s>}(b_k) = s_k$, $k \in [0, 7, 8, 15]$.

Remarquons que seul les plongements des demi-faces $<O_f>(b_0)$ et $<O_f>(b_8)$ sont modifiés par la règle de transformation. Les autres données de plongement sont conservées.

La règle de déconnexion munie des plongements ne suffit pas à modéliser l'étape 3. Nous devons nous assurer que la règle ne peut être appliquée que lorsque la vésicule filtrée contient suffisamment de protéines. Pour cela, nous attribuons la précondition de plongement suivante à la règle :

$$\text{Protéines}(p) > T$$

Nous supposons que T est une valeur entière choisie par l'utilisateur. La variable p fait référence à la variable p de L , après le filtrage. Enfin, $\text{Protéines}(p)$ est la quantité de protéines contenues dans la liste de protéines dénotée par p .

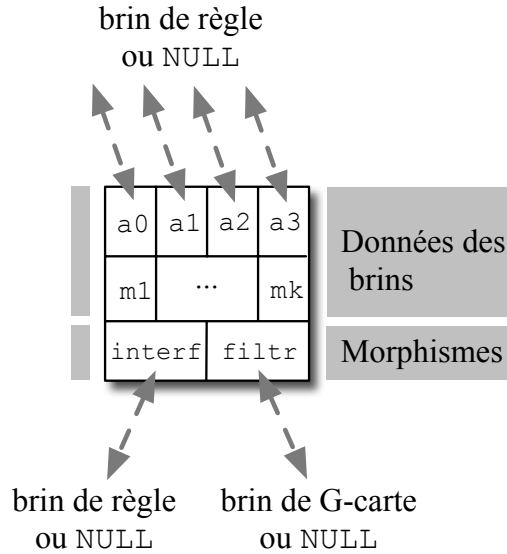


FIGURE 7.20 – Notre structure de brins de règles

Contrairement aux brins d'une 3-G-carte, les brins d'une règle de transformation de 3-G-carte n'ont pas tous leurs voisins par $\alpha_0, \alpha_1, \alpha_2$ et α_3 . En effet, sur la déconnexion, on remarque que les contraintes d'arcs incidents $\text{Incid}_L(\Sigma_E)$ et $\text{Incid}_R(\Sigma_E)$ (avec $\Sigma_E = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$) ne sont pas satisfaites. Par exemple, dans L et R , le brin b_0 ne possède pas de voisin par α_2 . La structure de données que nous proposons pour implanter les brins de règles tient compte de cette observation. Notre structure est illustrée en Fig. 7.20, on remarque que les pointeurs de brins de règles a_0, a_1, a_2 et a_3 peuvent avoir pour valeur NULL. Une structure de brins de règles dont le pointeur a_i a pour valeur NULL (avec $i \in [0, 3]$)

modélise un brin de règles qui ne possède pas de voisin par α_i .

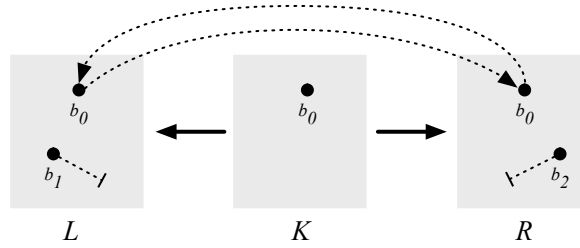


FIGURE 7.21 – Illustration du pointeur *interf*

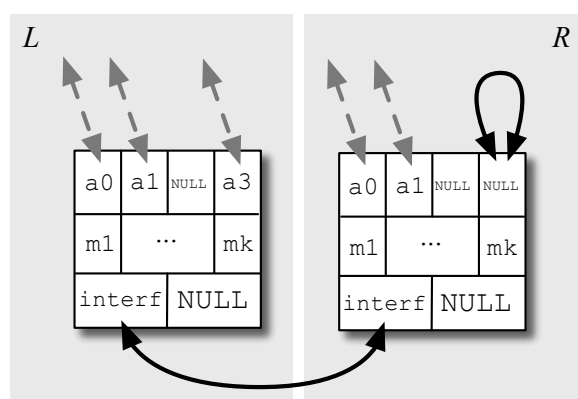
Nous modélisons l'interface *K* d'une règle de transformation par le pointeur de brins de règles *interf*. Nous implantons une règle de transformation par deux listes de brins – une pour *L* et l'autre pour *R* – munies de connexions entre les brins de *L* et ceux de *R*. Trois cas de connexion existent selon qu'un brin est conservé, ajouté ou supprimé par la règle (voir Fig. 7.21) :

- lorsqu'un brin b_0 de *L* pointe sur un brin de *R* par son pointeur *interf* (sur la figure, le pointeur *interf* est représenté par une flèche en pointillés), il s'agit d'un brin conservé, c'est-à-dire un brin de *K*. Dans ce cas, le pointeur *interf* est bidirectionnel ;
- lorsque le pointeur *interf* d'un brin b_1 de *L* a pour valeur NULL (sur la figure, le pointeur *interf* a valeur NULL est représenté par une flèche en pointillés terminée par un trait), il s'agit d'un brin supprimé, c'est-à-dire un brin de *L* qui n'apparaît ni dans *K* ni dans *R* ;
- lorsque le pointeur *interf* d'un brin b_2 de *R* a pour valeur NULL il s'agit d'un brin ajouté, c'est-à-dire un brin de *R* qui n'apparaît ni dans *K* ni dans *L*.

Notre structure de données établit également un lien entre les brins de règles et les brins de 3-G-cartes filtrés par la règle. Le pointeur *filtr* modélise ce lien. Intuitivement, lorsque la partie gauche d'une règle a été filtrée sur un graphe cible, chaque brin de règles de la partie gauche pointe sur le brin de G-cartes qu'il filtre dans le graphe cible. Les brins de règles de la partie droite possèdent également un pointeur *filtr*. Dans ce cas, il permet d'établir un lien entre les brins de règles et les brins de G-carte ajoutés par la règle. Nous reviendrons plus en détail sur l'utilisation du pointeur *filtr* lorsque, dans la section suivante, nous présenterons l'algorithme d'application d'une règle de transformation.

Remarquons enfin que comme les brins de 3-G-cartes, les brins de règles possèdent un jeu de marques entières m_1 à m_k . Celles-ci sont notamment utilisées lors du parcours des brins d'une règle. De même, nous associons à notre structure de brins de règles des opérations élémentaires analogues à celles introduites en section 7.2.2.2 dans le cas de des brins de 3-G-cartes. Remarquons toutefois que contrairement à l'opération d'ajout d'un brin de 3-G-cartes, l'opération d'ajout d'un brin de règles initialise les pointeurs a_0 , a_1 , a_2 et a_3 à NULL. De plus, nous proposons deux opérations qui permettent respectivement d'instancier les pointeurs *interf* et *filtr*.

La Fig. 7.22 illustre notre structure de brins de règles dans le cas de la règle de déconnexion. Deux structures de brin de règles apparaissent sur la Fig. 7.22. La structure de gauche modélise le brin b_0 de la partie gauche de la règle. À l'inverse, la structure de droite modélise le brin b_0 de la partie droite de la règle. Ainsi, bien que les morphismes d'inclusion de la règle de transformation nous indiquent que les brins b_0 des parties gauche et droite sont les mêmes, ils sont bien implantés par deux structures distinctes. Dans la partie gauche de la règle, b_0 n'a pas de voisin par α_2 . Par conséquent, le pointeur a_2 de la structure de gauche est à NULL. Dans la règle, on remarque que le brin b_0 apparaît dans les parties droite et gauche, de même que dans l'interface *K*. Ainsi, la structure de gauche est connectée à la structure de

FIGURE 7.22 – Implantation du brin de règles b_0 de la règle de déconnexion

droite par le pointeur `interf`.

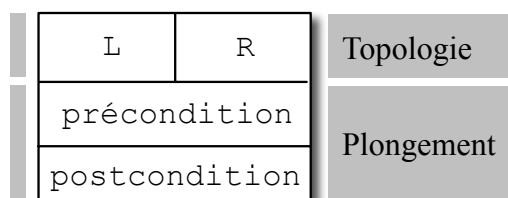


FIGURE 7.23 – Implantation d’une règle de transformation

Sur la Fig. 7.20, on remarque que les brins de règles ne portent pas de donnée de plongement. À terme, nous devons ajouter ces données aux différents champs d’un brin de règles. De cette manière, nous pourrions filtrer à la fois la topologie et le plongement *via* les brins de la partie gauche d’une règle. De plus, les données de plongements des brins de la partie droite permettront de mettre à jour facilement les plongements des brins de G-carte modifiés ou ajoutés par une règle. À l’heure actuelle, nous séparons les traitements de la topologie et du plongement. Le filtrage ainsi que la modification des plongements sont effectués par des fonctions `précondition` et `postcondition` écrites par l’utilisateur. Notre structure de données de règles de transformation, où interviennent ces deux fonctions, est illustrée en Fig. 7.23. Dans notre structure, `L` et `R` sont deux listes de brins de règles qui modélisent respectivement la partie gauche et la partie droite d’une règle. La fonction `précondition` effectue le filtrage des plongements, elle est appelée après le filtrage de la topologie et retourne un booléen qui indique si le plongement a pu être filtré. Elle prend en paramètre la partie gauche de sa règle associée. La fonction `postcondition` effectue les modifications de plongement lorsque la topologie et le plongement ont été filtrés, elle est appelée à la fin de l’algorithme d’application d’une règle de transformation.

La fonction `précondition` associée à la règle de déconnexion est donnée – à titre d’illustration – dans le Code 7.5. Rappelons que les données de plongements portées par les brins de règle de la partie gauche nous permettent de nous assurer que la demi-face $< O_f > (b_0)$ appartient à une vésicule. La fonction `précondition` proposée effectue ce filtrage. Les lignes 3 et 4 récupèrent le brin de règles b_0 ainsi que le brin de 3-G-cartes qu’il filtre. Remarquons l’utilisation du pointeur `filter` à la ligne 4. Les lignes 6 à 11 effectuent un filtrage `Caml`⁸. La ligne 10 est évaluée lorsque le brin de 3-G-cartes filtré

8. Le filtrage de `Caml`, *via* le mot-clé `match` permet de définir une expression par cas, en fonction de la valeur prise par la donnée filtrée.

```

1 let précondition L =
2   (* récupération des brins utiles pour le filtrage des plongements
3     *)
4   let brin_L_0 = List.nth L 0 in
5   let brin_filtré_0 = !^(brin_L_0.filtr) in
6
7   (* filtrage des plongements *)
8   match brin_filtré_0.plongement_volume with
9     Volume(Volume_Vésicule(p)) ->
10      (* test de la précondition de plongement *)
11      protéines(p) > T
12    | _ -> false

```

Code 7.5 – Précondition de plongement pour la règle de déconnexion

appartient à une vésicule, elle teste si la quantité de protéines de la vésicule est suffisante.

```

1 let postcondition R =
2   (* récupération des brins utiles pour les modifications des
3     plongements *)
4   let brin_R_0 = List.nth R 0 in
5   let brin_filtré_0 = !^(brin_R_0.interf.filtr) in
6   let brin_R_8 = List.nth R 8 in
7   let brin_filtré_8 = !^(brin_R_8.interf.filtr) in
8
9   (* modification des plongements *)
10  parcours_01
11    brin_filtré_0
12    (function b -> b.plongement_demi_face <- Demi_Face_Défaut) ;
13  parcours_01
14    brin_filtré_8
15    (function b ->
16      b.plongement_demi_face <- Demi_Face(Demi_Face_Bord)

```

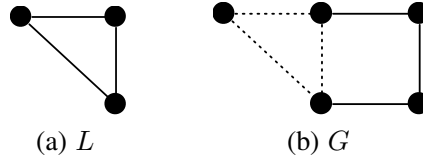
Code 7.6 – Postcondition de plongement pour la règle de déconnexion

La fonction `postcondition` du Code 7.6 effectue les modifications de plongement mises en jeu dans la règle de déconnexion. Les lignes 3 à 6 nous permettent de récupérer les brins de règles b_0 et b_8 ainsi que les brins de G-carte qu'ils filtrent. Dans les lignes 9 à 15, nous utilisons la fonction de parcours des demi-faces `parcours_01` – introduite en section 7.2.2.2 – afin de propager les modifications de plongement sur les brins des demi-faces $\langle O_f \rangle (b_0)$ et $\langle O_f \rangle (b_8)$. La demi-face $\langle O_f \rangle (b_0)$ est à présent de type `Demi_Face_Défaut` tandis que la demi-face $\langle O_f \rangle (b_8)$ devient une demi-face de type `Demi_Face_Bord`.

7.2.4.2 Application des règles de transformation

Dans cette section, nous introduisons le moteur d'application des règles de notre simulateur. Dans un premier temps, nous détaillons l'algorithme de filtrage de la topologie. Intuitivement, il s'agit de rechercher la partie gauche d'une règle dans un graphe cible. En fin de section, nous évoquons brièvement de quelle manière cet algorithme générique – il est le même pour toutes les règles – s'articule avec les fonctions `précondition` et `postcondition` développées par l'utilisateur pour chaque règle.

En théorie des graphes, la recherche de motifs dans des graphes est un problème récurrent [Zampelli 07]. Ce problème est illustré en Fig. 7.24. On se place dans le cas des graphes non étiquetés. Dans l'exemple, il s'agit de tester s'il existe un sous-graphe $G' = (V_{G'}, E_{G'}, s_{G'}, t_{G'})$ de $G = (V_G, E_G, s_G, t_G)$ (voir Fig. 7.24(a)) isomorphe à un graphe motif $L = (V_L, E_L, s_L, t_L)$ (voir Fig. 7.24(b)). Dans la pratique, on veut déterminer s'il existe un sous-graphe G' de G tel que l'on puisse construire un isomorphisme $\varphi : L \rightarrow G'$. Dans l'exemple, un tel sous-graphe G' existe, il est représenté en pointillé sur la Fig. 7.24(b). On remarque que le problème de la recherche de L dans G revient à regarder si l'on peut « superposer » L à G . Dans le cas général, ce problème est NP-difficile.

FIGURE 7.24 – Deux graphes L et G

Donnons-nous à présent une contrainte sur l'isomorphisme φ . On se donne un sommet u de V_L et un sommet v de $V_G' \subset V_G$ et nous imposons que φ soit tel que $\varphi(u) = v$. Étudions l'algorithme de recherche d'isomorphisme à la lumière de cette nouvelle contrainte. Intuitivement, l'algorithme est basé sur un parcours en parallèle – avec marquage – des sommets de L et de G . Prenons l'exemple de la Fig. 7.25(a). Nous débutons le parcours en marquant les sommets u et v à 1 (sur la figure, le marquage des sommets u et v est implicite). Il s'agit ensuite de choisir un voisin de u puis un voisin de v et de les marquer à 2. La valeur de la marque augmente au fur et à mesure du parcours ; elle peut être vue comme une date de passage sur les noeuds. La recherche réussie lorsque tous les sommets du motif sont marqués et l'isomorphisme φ (représenté par les flèches en pointillés) est donné par les marques. Au cours de l'algorithme, un marquage incohérent permet de détecter l'échec de la recherche. Nous ne souhaitons pas rentrer outre mesure dans le détail de l'algorithme. Retenons simplement que plus la connexité de L et de G est élevée, plus il existe de parcours possibles et donc moins l'algorithme est efficace. La Fig. 7.25 illustre quatre marquages possibles des sommets de L et de G . Dans les cas illustrés par les Fig. 7.25(a) – que nous venons d'étudier – et 7.25(b), la recherche réussie. Les Fig. 7.25(c) et 7.25(d) montrent deux marquages qui ne permettent pas de construire φ . Dans ce cas, il faut effectuer un retour sur trace – ou « backtracking » – dans l'algorithme.

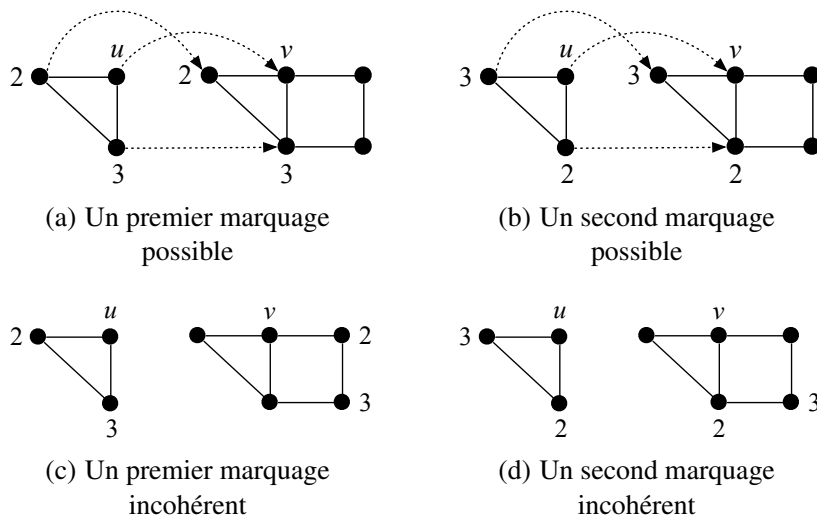


FIGURE 7.25 – Illustration de l'algorithme de recherche de motifs

Donnons-nous à présent une contrainte sur les arêtes de L et G . Nous supposons à présent que les arêtes sont étiquetées sur un ensemble Σ_E et qu'un sommet ne peut pas avoir deux arêtes incidentes qui portent la même étiquette. Nous nous plaçons alors dans la catégorie $\mathbb{G}e(\emptyset, \Sigma_E)$ des graphes partiellement étiquetés (on suppose que les sommets ne sont pas étiquetés). Dans ce cas, l'isomorphisme φ conserve l'étiquetage des arêtes (la conservation de l'étiquetage des sommets est triviale). Prenons pour exemple les graphes $L = (V_L, E_L, s_L, t_L, l_{V_L}, l_{E_L})$ et $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$, illustrés respectivement en Fig. 7.26(a) et 7.26(b). Comme précédemment, le parcours débute sur les sommets v et u . Comme l'isomorphisme φ conserve les étiquettes des arêtes, le choix d'un sommet voisin dans L (respectivement G) détermine le choix du sommet voisin dans G (respectivement L). Ainsi, si nous poursuivons le parcours de L avec le sommet u' connecté à u par a , nous devons choisir – dans G – le sommet v' connecté à v par a . Pour peu que l'on définisse un ordre – avec une priorité sous-jacente – sur les étiquettes des arêtes, l'algorithme devient linéaire en le nombre de noeuds de L puisqu'à chaque étape du parcours, il existe un unique voisin possible. C'est cette version de l'algorithme que nous avons implantée dans le prototype de simulateur.

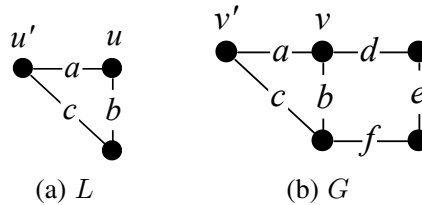


FIGURE 7.26 – Deux graphes L et G étiquetés

En effet, une 3-G-carte est un graphe $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ étiqueté sur (Σ_V, Σ_E) – avec Σ_V un ensemble d'étiquettes de sommets et $\Sigma_E = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$ un ensemble d'étiquettes d'arêtes – qui satisfait la contrainte $\text{Incede}_G(\Sigma_E)$. De même, la partie gauche d'une règle est un graphe $L = (V_L, E_L, s_L, t_L, l_{V_L}, l_{E_L})$ étiqueté sur $(\Sigma_V, \Sigma'_E \subset \Sigma_E)$ qui satisfait la contrainte $\text{Incede}_L(\Sigma'_E)$. Ainsi, aucun sommet de V_G ou V_L ne peut avoir deux arêtes incidentes qui portent la même étiquette. Pour l'heure, nous n'avons pas implanté le filtrage du plongement de manière générique, il est effectué à l'aide de la fonction `précondition` développée par l'utilisateur pour chaque règle. Nous considérons alors que lors du filtrage, les brins de V_G et de V_L portent l'étiquette \perp . Notre algorithme de filtrage de la topologie, qui recherche un motif L dans G à partir d'un brin u de V_L et d'un brin v de V_G , est implanté dans la fonction `filtrage` du Code. 7.7.

Lors du parcours, le marquage est effectué avec le pointeur d'entier `date`. Il est initialisé à la ligne 3. Le parcours est effectué par la fonction auxiliaire récursive `filtrage_rec` (lignes 6 à 30). La ligne 8 nous indique que si, lors du parcours, les brins courants ne présentent pas la même marque `m1`, on lève l'exception `Filtrage_impossible` et la fonction `filtrage` retourne `false` (le traitement de l'exception est effectué dans les lignes 34 à 38). Ce cas d'erreur est illustré en Fig. 7.27. On cherche à filtrer la demi-face triangulaire L de la Fig. 7.27(a) dans la 3-G-carte G de la Fig. 7.27(b). Avant l'appel de la fonction `filtrage`, les brins de L et G portent la marque 0. Le parcours débute sur les brins u de L et v de G , que l'on marque à 1. Au cours du parcours, nous sommes amenés à appeler récursivement la fonction `filtrage_rec` avec en paramètre les brins situés dans les régions grisées. Le brin grisé de L porte la marque implicite 1 tandis que le brin grisé de G – qui n'a pas encore été traité – porte la marque 0. Le test de la ligne 8 réussit et l'exception `Filtrage_impossible` est levée. Il n'existe pas de sous-graphe isomorphe à L dans G .

Dans la fonction `filtrage_rec`, lorsque les brins courants n'ont pas été traités (voir le test de la ligne 11), on exécute le bloc qui débute à la ligne 12 et termine à la ligne 28. Les marques sont mises à jour dans les lignes 12 à 16. À la ligne 18, on connecte le brin courant du motif au brin qu'il filtre dans

```

1 let filtrage u v =
2   (* initialisation de la marque *)
3   let date = ref 0 in
4
5   (* fonction auxiliaire récursive *)
6   let rec filtrage_rec u v =
7     (* cas de filtrage impossible *)
8     if (u.ml <> v.ml) then raise Filtrage_impossible ;
9
10    (* si les brins courants ne sont pas marqués alors on continue *)
11    if ((u.ml = 0) && (v.ml = 0)) then (
12      (* mise à jour de la date *)
13      date := !date + 1 ;
14      (* marquage des brins courants *)
15      u.ml <- !date ;
16      v.ml <- !date ;
17      (* connexion du brin de règle u au brin filtré v *)
18      u.filtr <- nouveau_pointeur v ;
19
20      (* appel récursif sur les voisins *)
21      if (u.a0 != Null) then (* si u a un voisin par a0 *)
22        filtrage_rec !^(u.a0) !^(v.a0) ;
23      if (u.a1 != Null) then (* si u a un voisin par a1 *)
24        filtrage_rec !^(u.a1) !^(v.a1) ;
25      if (u.a2 != Null) then (* si u a un voisin par a2 *)
26        filtrage_rec !^(u.a2) !^(v.a2) ;
27      if (u.a3 != Null) then (* si u a un voisin par a3 *)
28        filtrage_rec !^(u.a3) !^(v.a3) ;
29    )
30  in
31
32  (* si Filtrage_impossible n'a pas été levé alors le filtrage réussit *)
33
34  try (
35    (* appel de la fonction auxiliaire récursive *)
36    filtragerec u b ;
37    true ;
38  ) with Filtrage_impossible -> false

```

Code 7.7 – Notre algorithme de filtrage de la topologie

le 3-G-carte *via* son pointeur `filtr`. Intuitivement, c'est de cette manière que l'on construit l'isomorphisme φ . Lorsqu'un motif est reconnu, tous les brins du motif sont connectés aux brins qu'ils filtrent dans la 3-G-carte cible. Dans les lignes 20 à 28, la fonction `filtrage_rec` est appelée récursivement sur les voisins des brins courants. Les tests des lignes 21, 23, 25 et 27 nous assurent de ne pas suivre dans G , les α_i (avec $0 \leq i \leq n$) qui n'existent pas dans L . Par exemple, sur la Fig. 7.27, les brins de G sont parcourus en utilisant uniquement les liaisons qui existent dans L . Enfin, on remarque que les lignes 22, 24, 26 et 28 nous assurent que le parcours est effectué en utilisant simultanément les mêmes liaisons dans le motif et dans la 3G-carte.

Deux cas de réussite du filtrage sont illustrés en Fig. 7.28. Le premier cas est illustré par les Fig. 7.28(a) et 7.28(b). Le second est illustré par les Fig. 7.28(c) et 7.28(d). Seul le choix des sommets u et v varie d'un

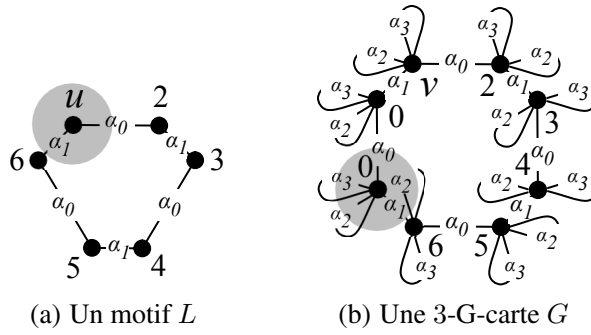


FIGURE 7.27 – Illustration du cas d'erreur de la fonction `filtrage_rec`

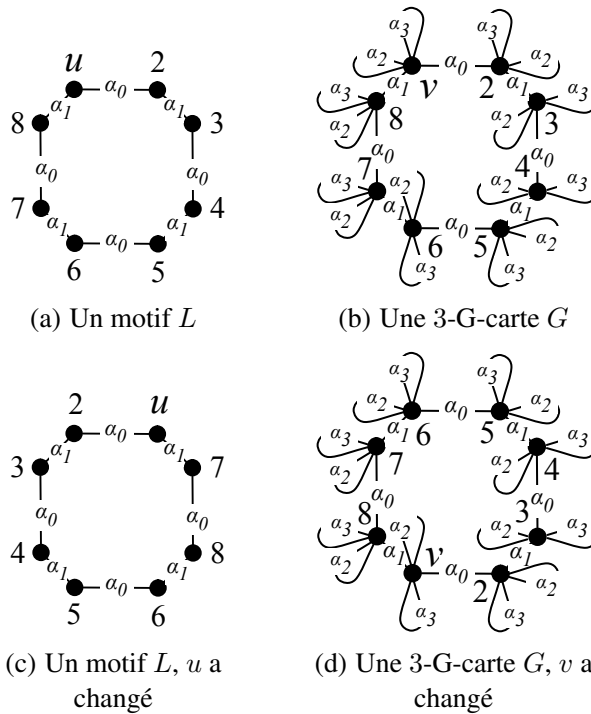


FIGURE 7.28 – Deux filtrages réussis

cas à l'autre. Dans les deux cas, le parcours termine sans que l'exception `Filtrage_impossible` soit levé et l'isomorphisme φ est donné par les dates qui étiquettent les brins. Cet exemple montre que parfois, le choix des sommets u et v importe peu. Ce choix détermine φ mais n'influe pas systématiquement sur l'échec ou la réussite du filtrage.

La fonction d'application d'une règle à une 3-G-carte est introduite dans le Code 7.8. Nous l'illustrons en Fig. 7.29. Nous décomposons l'application d'une règle en cinq étapes. L'étape 1 (lignes 2 à 7) est illustré en Fig. 7.29(a). Nous souhaitons appliquer la règle de déconnexion de la Fig. 7.19 page 186 à un coin de la face qui connecte une vésicule à un saccule. L'étape 1 met en oeuvre le filtrage de la topologie *via* la fonction `filtrage_global`. Intuitivement, `filtrage_global` est analogue à la fonction `filtrage` du Code 7.7 à ceci près qu'elle permet de filtrer un motif composé de plusieurs composantes connexes. À l'issue de l'étape 1, les brins de L sont connectés aux brins filtrés par leur pointeur `filtr`, représenté par une flèche épaisse. Pour le moment, les pointeurs `filtr` des brins de R sont à `NULL`, ils sont également représentés par des traits épais. Enfin, les flèches en pointillés illustrent

```

1  let application règle gcarte =
2    (* ÉTAPE 1 : filtrage de toutes les composantes connexes de la
3      règle *)
4    if (not (filtrage_global règle.L gcarte)) then
5      raise Application_impossible ;
6    (* les brins de règle.L sont connectés au brin qu'ils filtrent par
7      leur pointeur filtr *)
8
9    (* ÉTAPE 2 : filtrage des plongements *)
10   if (not (règle.précondition règle.L)) then
11     raise Application_impossible ;
12
13   (* ÉTAPE 3 : traitement des brins à ajouter et à supprimer *)
14   ajout_suppression règle gcarte ;
15   (* les brins règle.R sont tous connectés à un brin dans gcarte par
16     leur pointeur filtr *)
17
18   (* ÉTAPE 4 : traitement des modifications topologiques *)
19   List.iter
20     (fun brin_R ->
21       (* traitement de a0 *)
22       if (brin_R.a0 <> Null) then (
23         let brin_filtré = (!^(brin_R.filtr)) in
24         let nouveau_voisin_brin_filtré =
25           (!^((!^(brin_R.a0)).filtr)) in
26           connexion 0 brin_filtré nouveau_voisin_brin_filtré)
27       (* traitements analogues pour a1, a2 et a3 *)
28       ... )
29   règle.R ;
30
31   (* ÉTAPE 5 : traitement des modifications de plongements *)
32   règle.postcondition règle.R

```

Code 7.8 – Notre fonction d'application d'une règle à une 3-G-carte

la connexion des brins de L et R par leur pointeur `interf`.

Lors de l'étape 2 (lignes 9 à 11), la fonction `précondition` de la règle de déconnexion – introduite dans le Code. 7.5 page 190 – est appelée. Elle permet de filtrer le plongement et de tester la condition de plongement. À ce stade, les demi-faces auxquelles appartiennent les quatre brins de G sont virtuelles.

L'étape 3 (lignes 13 à 16) est illustrée par la Fig. 7.29(b). Afin de clarifier la figure, nous ne faisons plus apparaître les pointeurs des brins de L . Lors de cet étape, chaque brin ajouté – c'est-à-dire ceux qui ont leur pointeur `interf` à NULL dans R – est connecté à un nouveau brin dans G par son pointeur `filtr`. De plus, les brins de G filtrés par des brins supprimés – c'est-à-dire ceux qui ont leur pointeur `interf` à NULL dans L – sont retirés de G . Enfin, les brins conservés sont connectés aux brins de R par leur pointeur `filtr`. Dans notre exemple, comme la règle de déconnexion n'ajoute pas ni ne supprime de brin, l'étape 3 établit simplement la connexion de R vers G . Dans la pratique, elle est construite en utilisant les pointeurs `interf` qui, sur la Fig. 7.29(a), lient indirectement les brins de R à G .

Les modifications de la topologie mises en jeu dans la règle de déconnexion sont effectuées lors de l'étape 4 (lignes 18 à 29). Celle-ci est illustrée en Fig. 7.29(c). On remarque que dans G , les liens α_3

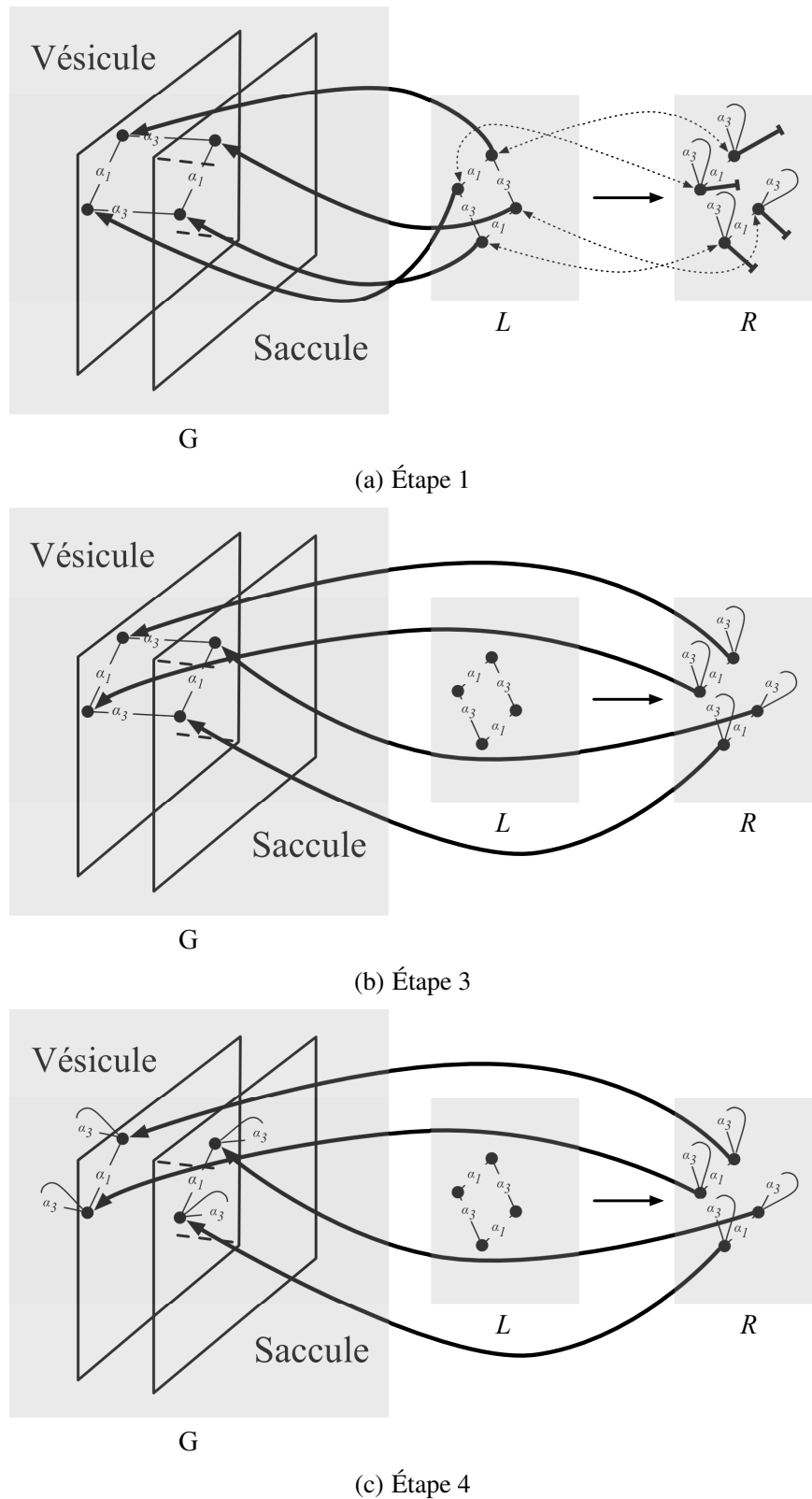


FIGURE 7.29 – Illustration de l'application d'une règle de transformation

deviennent des boucles. Enfin, à l'étape 5 (lignes 31 et 32), un appel à la fonction `postcondition` du Code 7.6 page 190 permet d'effectuer les modifications de plongements. À l'issue de cette étape, la

demi-face virtuelle de la vésicule devient une demi-face de type `Demi_Face_Défaut` et la demi-face virtuelle du saccule devient une demi-face de type `Demi_Face_Bord`.

7.2.5 Bilan et perspectives

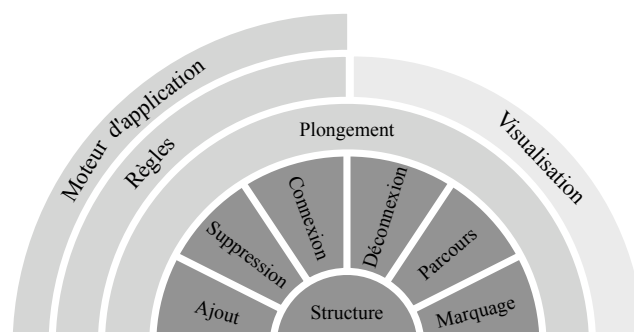


FIGURE 7.30 – Architecture du simulateur

Dans ce chapitre, nous avons présenté notre premier prototype de simulateur biologique à base topologique. La Fig. 7.30 illustre l'ensemble des composantes du simulateur, tel qu'il existe aujourd'hui. Dans la section 7.2.2, nous avons introduit notre noyau de simulateur (il apparaît en gris foncé sur la figure), il est basé sur une structure de brins de 3-G-cartes munie d'un jeu d'opérations de base. En section 7.2.3, nous avons introduit les structures de données qui implantent les plongements de nos objets. Rappelons que dans notre simulateur, nous plongeons les volumes, les demi-faces, les arêtes simples et les 0-cellules. Dans cette même section, nous avons présenté les différents modes de visualisation du simulateur. Nous avons vu que les visualisations s'appuient principalement sur les plongements. Enfin, dans la section 7.2.4, nous avons introduit une structure de données qui plante les règles de transformation de 3-G-cartes sans variable. Nous avons également présenté notre moteur d'application des règles en étudiant notamment les mécanismes à la base du filtrage de la topologie.

Dans cette section, nous évoquons principalement deux perspectives à la lumière desquelles nous devons poursuivre le développement du simulateur. Dans un premier temps, nous étudions brièvement l'influence de la stratégie d'application des règles sur les phénomènes qui émergent d'une simulation. Dans un second temps, nous évoquons l'implantation future des règles avec variables.

De l'importance de la stratégie d'application des règles

En l'état, notre simulateur ne propose pas de stratégie préconstruite pour l'application des règles de transformation. Ainsi, l'utilisateur doit écrire sa propre stratégie en spécifiant notamment dans quel ordre sont appliquées les règles et également l'ordre dans lequel une règle donnée est appliquée aux objets. On distingue principalement deux types de stratégies, des *stratégies déterministes* et des *stratégies stochastiques*. Notons que la stochasticité peut se situer à plusieurs niveaux. L'ordre dans lequel sont appliquées les règles peut être aléatoire. De plus, pour une règle donnée, on peut choisir aléatoirement sur quelle partie d'objet on essaye de l'appliquer. Dans tout les cas, le choix de la stratégie influe sensiblement sur les phénomènes qui émergent d'une simulation. La Fig. 7.31 illustre cette observation.

Ici, il s'agit d'étudier l'influence de la stratégie sur le transport de protéines solubles dans un modèle de l'appareil de Golgi connecté. Les Fig. 7.31(a), 7.31(b) et 7.31(c) montrent respectivement l'état du modèle au temps t_1 , t_2 et t_3 dans le cas d'une stratégie purement déterministe. Ainsi, à chaque pas de simulation, les règles sont appliquées dans le même ordre et les parties d'objets ciblées par les règles sont également choisies dans le même ordre. Les Fig. 7.31(d), 7.31(e) et 7.31(f) montrent l'objet au même

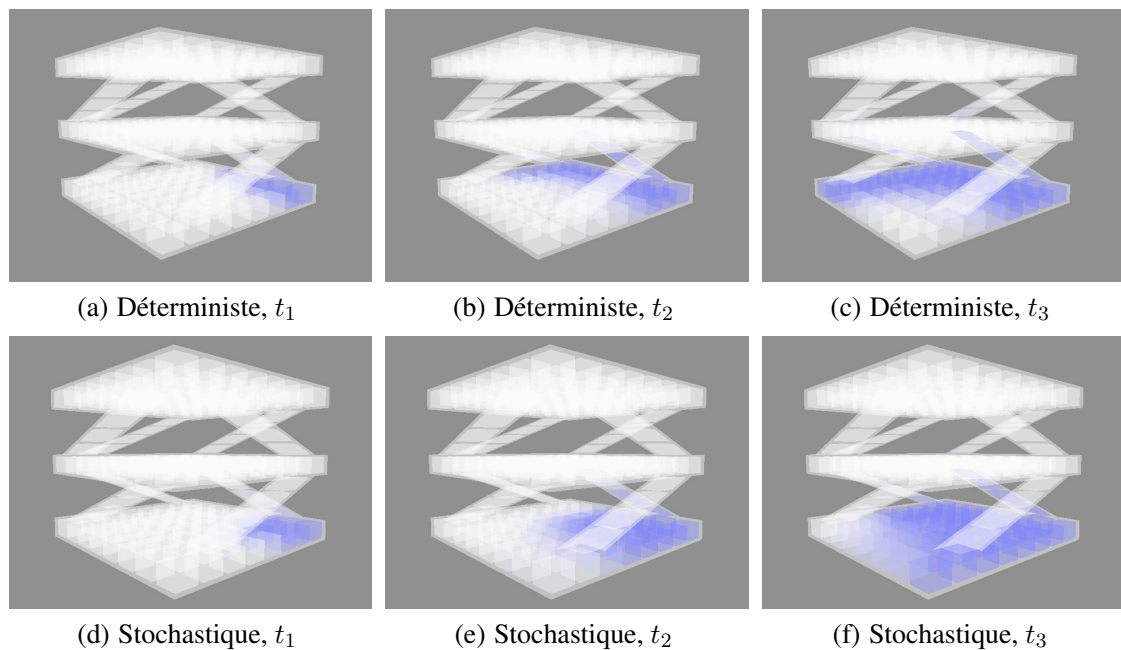


FIGURE 7.31 – Comparaisons des stratégies déterministe et stochastique

instant dans le cas d’une stratégie où les parties d’objets sont sélectionnées aléatoirement. On remarque que le comportement des protéines diffère sensiblement d’une stratégie à une autre. Avec la stratégie déterministe, les protéines progressent dans une direction privilégiée induite par l’ordre dans lequel les parties d’objets sont sélectionnées. En particulier, les protéines « évitent » le centre du saccule *cis* (le saccule du bas sur les figures). Dans le cas de la stratégie stochastique, les protéines sont transportées de manière uniforme dans toutes les directions. D’après les biologistes, cette stratégie apparaît plus réaliste.

```

1 type t_protéine = {
2   anciennes : int ;
3   nouvelles : int
4 }

```

Code 7.9 – Notre implantation des protéines

Dans le simulateur final, nous devons proposer à l’utilisateur des squelettes de stratégies de manière à automatiser au maximum la mise en place d’une simulation. Remarquons que le développement de telles stratégies préconstruites doit être mené avec vigilance si l’on souhaite préserver au maximum le réalisme des simulations. La gestion des protéines compte parmi les points délicats à traiter. En effet, il faut notamment veiller à ce qu’une protéine donnée ne puisse être transportée deux fois au cours d’une itération. Nous avons implanté un type Caml `t_protéine` qui permet d’éviter ces artefacts, nous l’introduisons dans le Code. 7.9.

Dans la section 7.2.3, nous avons montré comment associer des protéines – en fait des valeur de type Caml `t_protéine` – aux compartiments et membranes biologiques. Une valeur de type Caml `t_protéine` comprend deux champs. L’entier `anciennes` modélise les protéines disponibles dans un compartiment, ou dans une membrane, au début d’une itération. Lorsqu’une protéine parmi les disponibles est transportée dans un second compartiment, la valeur du champ `anciennes` est décrémentée. En parallèle, le champ `nouvelles` du second compartiment est incrémenté. De cette manière, nous sommes assurés qu’une protéine ne sera pas transportée deux fois, ou pire encore dédoublée, au court d’un pas de simulation. À la fin de l’itération, les champs `anciennes` sont réinitialisés avec leur champ

nouvelle associé et les champs nouvelles sont remis à zéro. Ainsi, les protéines transportées au cours d'une itération sont à nouveau disponibles à l'itération suivante. Les protéines de notre premier prototype de simulateur sont implantées de cette manière.

Implantation des règles avec variables

Les règles de transformation avec variables ne sont pas implantées dans notre premier prototype de simulateur. Dans les premières simulations que nous avons développées, nous avons pu nous passer des règles avec variables en raison de l'homogénéité des objets manipulés. Notamment, dans les modèles du transport et du bourgeonnement, nous ne manipulons que des faces carrées. Lorsque nous souhaiterons manipuler des modèles plus complexes tels que la tour présentée en section 6.5.2, le passage aux règles avec variables sera inévitable. En plus du développement des règles avec renommages d'orbites introduites dans la première partie du manuscrit, l'étude de modèles biologiques nous a mené à imaginer une nouvelle classe de règles avec variables, que nous devons également implanter.

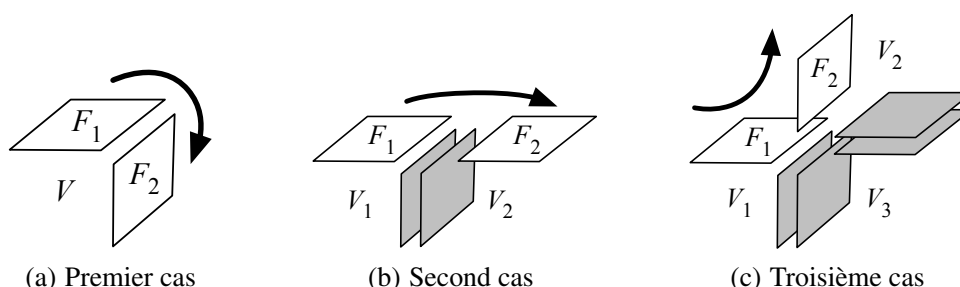


FIGURE 7.32 – Trois cas de transport membranaire

Dans la Fig. 7.32, nous étudions le transport des protéines membranaires depuis une portion de membrane F_1 vers une portion de membrane F_2 , toutes deux représentées par des demi-faces. Sur la Fig. 7.32(a), F_1 et F_2 appartiennent à un même volume et sont reliées par α_2 (les liens n'apparaissent pas sur la figure). Dans le second cas – Fig. 7.32(b) – F_1 et F_2 bordent deux volumes distincts reliés par deux demi-faces virtuelles (représentées en gris sur la figure). Enfin, dans le dernier cas – Fig. 7.32(c) – F_1 et F_2 appartiennent à nouveau à deux volumes distincts V_1 et V_2 , mais cette fois-ci, les volumes sont séparés par un troisième volume V_3 . Le volume V_3 est connecté à V_1 et V_2 par l'intermédiaire de deux couples de demi-faces virtuelles.

Dans chacun des cas, nous souhaitons transporter des protéines depuis F_1 vers F_2 . Cependant, nous ne pouvons effectuer ce transport *via* une seule et même règle de transport. Dans les faits, nous devons écrire une règle par cas. Ajoutons que la Fig. 7.32 n'est pas exhaustive, d'autres configurations existent en fonction du nombre de volumes qui séparent V_1 de V_2 . Afin de rassembler les différentes configurations en une seule et même règle avec variables, nous devons imaginer un nouveau type de variables permettant de filtrer une répétition de motifs, dans notre cas : une séquence de doubles demi-faces virtuelles. Nous souhaitons étudier ces nouvelles règles en nous appuyant sur la notion d'*expressions régulières*, où l'on sait répéter des motifs dans des mots.

Enfin, lorsque les règles avec renommages d'orbites – de même que cette première extension – seront implantées, nous devons mettre à disposition de l'utilisateur une interface intuitive pour l'écriture de ses règles. En ce sens, les langages biochimiques mentionnés en section 7.1.2 sont intéressants à étudier puisqu'ils proposent un langage proche des notations utilisées par les biologistes.

BILAN ET PERSPECTIVES

Transformations de graphes pour les opérations topologiques en modélisation géométrique

Dans ce manuscrit, nous avons cherché à exprimer les opérations topologiques de la modélisation géométrique par des règles de transformation de graphes dans l'objectif de nous rapprocher des langages à base de règles utilisés pour simuler ou animer des structures. Nous avons choisi les cartes généralisées comme modèle topologique. Dans la littérature, les cartes généralisées sont définies par une structure algébrique munie de contraintes qui assurent la cohérence topologique des objets. Du fait de la régularité de la structure des cartes généralisées, nous avons proposé de les définir comme de simples graphes partiellement étiquetés – et non orientés – dans lesquels les sommets représentent les éléments de base d'une carte généralisée, tandis que les arcs modélisent les relations de voisinage qui existent entre ces éléments. De fait, nous avons exprimé les contraintes de cohérence des cartes généralisées par deux contraintes de graphe : la contrainte d'arcs incidents et la contrainte de cycles. Nous avons ainsi pu définir les opérations topologiques des cartes généralisées comme étant des règles de transformation de graphes. Parmi les différentes approches des transformations de graphes, nous avons choisi l'approche par double somme amalgamée. De plus, afin de capturer au mieux la généricité des opérations topologiques définies sur les cartes généralisées, nous avons muni nos règles de nouvelles variables spécialisées.

Les opérations topologiques des cartes généralisées sont souvent génériques en fonction des orbites – c'est-à-dire des sous-graphes – considérées. Par exemple, afin de coller deux volumes le long d'une face triangulaire ou bien le long d'une face carrée, nous utilisons la même opération de 3-couture. Ainsi, dans le souci d'exprimer ces opérations génériques par des règles de transformations de graphes, nous avons été amenés à paramétrer nos règles par des variables spécifiques – nous les appelons les variables orbites – qui dénotent des familles d'orbites. Nos variables orbites sont originales dans la communauté des transformations de graphes. Elles s'apparentent aux variables clones définies par Hoffmann [Hoffmann 05]. En effet, lors de leur substitution, la re-connexion au contexte est effectuée automatiquement. Cependant, à la manière des variables graphes également définies par Hoffmann, elles permettent d'abstraire une importante variété de sous-graphes. À la différence des variables clones et graphes, nos variables sont typées, c'est-à-dire qu'elles dénotent des orbites typées par les étiquettes de leurs arcs. De plus, nous avons introduit des fonctions de renommage des variables qui permettent à la fois de renommer et de supprimer des arcs dans des orbites. Ces mécanismes de renommage sont essentiels dans de nombreuses opérations topologiques telles que les opérations de cône ou de chanfreinage. La règle p_{cone} qui traduit l'opération de cône des cartes généralisées de dimension 3 est illustrée en Fig. 7.33(a). Cette règle est munie d'une unique variable orbite – implicite – et de quatre fonctions de renommage $\langle \alpha_0\alpha_1 \rangle$, $\langle \alpha_0_ \rangle$, $\langle _ \alpha_2 \rangle$ et $\langle \alpha_1\alpha_2 \rangle$. En instanciant la variable orbite de p_{cone} par l'orbite O de type $\langle \alpha_0\alpha_1 \rangle$ (voir Fig. 7.33(b)) ou bien par l'orbite O' de type $\langle \alpha_1\alpha_2 \rangle$ (voir Fig. 7.33(c)), nous obtenons la même règle de transformation de graphes $p_{cone}^O = p_{cone}^{O'}$ illustrée en Fig. 7.33(d). Les fonctions de renommage de p_{cone} sont ainsi spécialisées au moment de l'instanciation. Par exemple, en instanciant par O , le renommage $\langle \alpha_0\alpha_1 \rangle$ est vu comme le renommage identité $\langle \alpha_0\alpha_1 \rangle : \langle \alpha_0\alpha_1 \rangle \rightarrow \langle \alpha_0\alpha_1 \rangle$. À l'inverse,

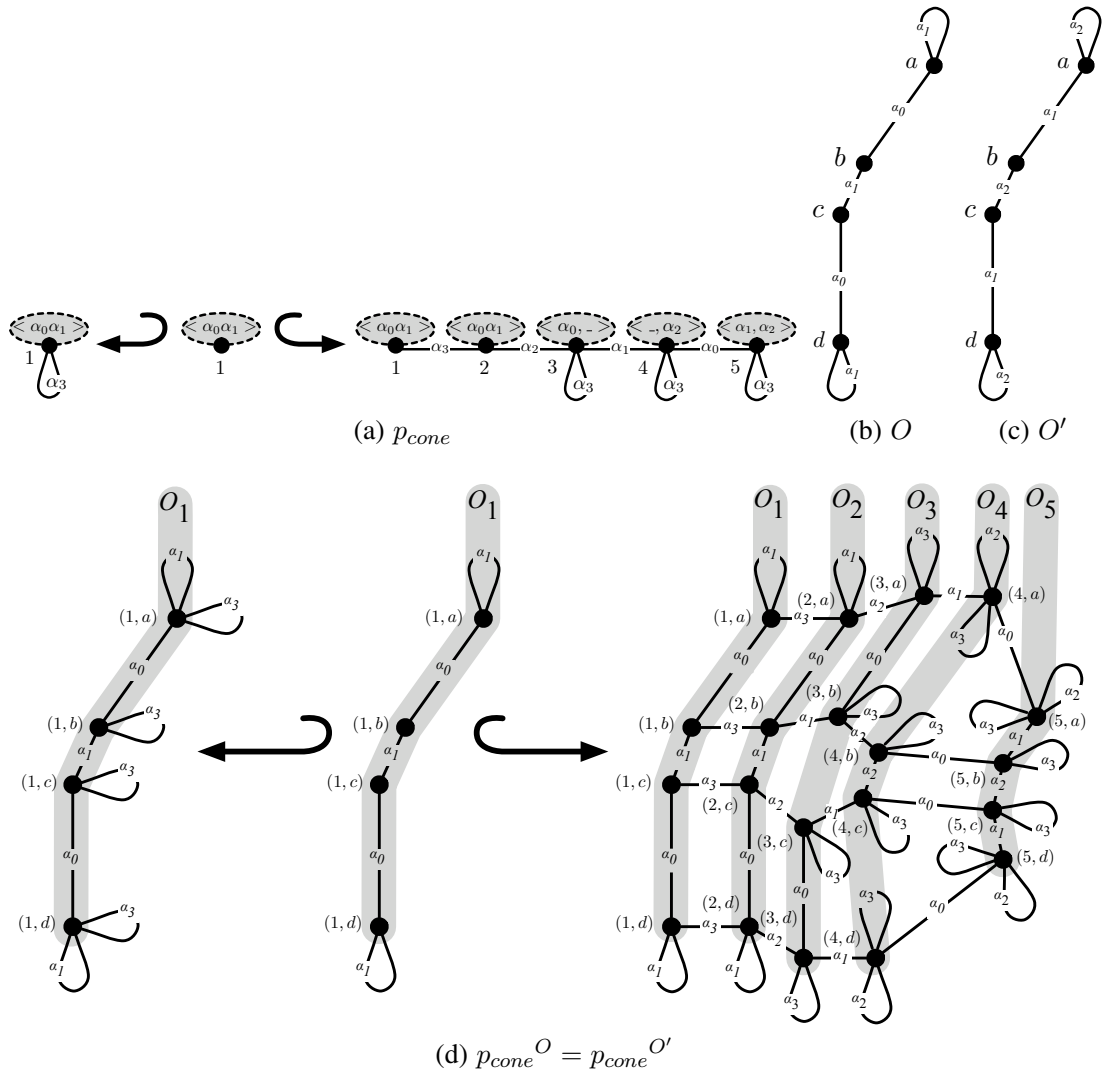


FIGURE 7.33 – La règle de cône

en instanciant par O' , ce même renommage est spécialisé en $\langle \alpha_0 \alpha_1 \rangle : \langle \alpha_1 \alpha_2 \rangle \rightarrow \langle \alpha_0 \alpha_1 \rangle$. Nous avons ainsi été conduits à introduire un mécanisme d'instanciation automatique, basé sur les produits fibrés. En utilisant des orbites de type $\langle \alpha_0 \alpha_1 \rangle$ mais de structures différentes (en particulier de tailles différentes), ce mécanisme nous permet d'instancier nos règles en toute une famille de règles de transformation de graphes classiques. Notre approche constitue une première étude de la combinaison des produits fibrés et des sommes amalgamées.

Afin de nous assurer que nos règles munies de renommages d'orbites préservent les contraintes de cohérence des cartes généralisées, nous avons introduit différents critères syntaxiques. Rappelons tout d'abord que nous définissons les cartes généralisées comme des graphes non orientés, c'est-à-dire des graphes dans lesquels les sommets sont connectés par des arêtes (une arête est vue comme deux arcs inversement orientés et munis de la même étiquette). Ainsi, un premier critère syntaxique nous assure que lorsqu'une règle est non orientée – c'est-à-dire lorsque sa partie gauche, son interface située au milieu et sa partie droite sont tous des graphes non orientés – et que nous l'appliquons à un graphe non orienté, le résultat de l'application est également un graphe non orienté. Dans une carte généralisée de dimension n , la contrainte d'arcs incidents nous indique que chaque sommet possède exactement $n + 1$ arcs incidents étiquetés par α_0 à α_n . Ainsi, dans nos règles, les sommets supprimés et ajoutés lors de

l'application d'une transformation doivent être munis de tous leur arcs incidents. Par exemple, dans la partie droite de p_{cone} , le sommet 2 – qui abstrait l'orbite O_2 de p_{cone}^O – possède bien tous ses liens : il est muni de deux arêtes incidents étiquetés par α_2 et α_3 et les étiquettes α_0 et α_1 figurent dans son renommage $\langle \alpha_0 \alpha_1 \rangle$. On remarque ainsi qu'après instanciation, les sommets de l'orbite O_2 satisfont tous la contrainte d'arcs incidents. Notons qu'une condition analogue concerne également les sommets conservés, c'est-à-dire les sommets de l'interface de p_{cone} . Nous avons vu que la satisfaction de ce premier critère syntaxique assure également la satisfaction de la condition d'arcs pendants, qui dans les transformations de graphes, nous assurent qu'une règle peut être appliquée. Enfin, dans une carte généralisée, la contrainte de cycles impose la présence de cycles particuliers. Par exemple, dans une carte généralisée de dimension 3, chaque sommet est la source d'un cycle $\alpha_0 \alpha_2 \alpha_0 \alpha_2$. Un second critère syntaxique assure la satisfaction de la contrainte de cycles. Dans la partie droite de p_{cone} , on remarque notamment que les sommets 4 et 5 sont connectés par une arête étiquetée par α_0 et présentent tous les deux une fonction de renommage – respectivement $\langle _ \alpha_2 \rangle$ et $\langle \alpha_1 \alpha_2 \rangle$ – dans laquelle l'étiquette α_2 intervient à la même position. Il en résulte que dans p_{cone}^O , les sommets des orbites 0_4 et 0_5 sont tous la source d'un cycle étiqueté par $\alpha_0 \alpha_2 \alpha_0 \alpha_2$.

Nous avons ainsi défini un langage qui permet d'exprimer les opérations topologiques des cartes généralisées sous la forme de règles de transformation de graphes munies de variables. Ce langage se rapproche ainsi des langages à base de règles utilisés pour la simulation ou l'animation de systèmes structurés. Notre langage présente un avantage majeur par rapport à l'approche classique des opérations en modélisation géométrique. En effet, nos critères syntaxiques – qui assurent par construction le bien-fondé des opérations définies sous la forme de règles – nous évitent de prouver au cas par cas la cohérence topologique de chaque opération nouvellement définie. Ici, la simple écriture d'une opération sous la forme d'une règle assure la cohérence. Tout comme les précédents travaux établissant un lien entre les méthodes formelles et la modélisation géométrique [Bertrand 92, Ledoux 02a, Ledoux 02b], notre première étude de l'utilisation des transformations de graphes en modélisation géométrique nous permet d'automatiser en partie le travail de conception nécessaire au développement des opérations topologiques des modélisateurs géométriques.

Application à l'étude de la dynamique de l'appareil de Golgi

Dans ce manuscrit, nous avons étudié les bénéfices d'une étude de la topologie dans le cadre de la modélisation de systèmes biologiques cellulaires. Parmi les nombreuses organelles de la cellule, l'appareil de Golgi joue un rôle important dans le transport des protéines depuis la périphérie du noyau jusqu'à l'environnement extra-cellulaire. De nombreuses images issues de la microscopie électronique suggèrent que l'appareil de Golgi est constitué d'une pile de citernes déconnectées et entourées de vésicules. A partir de ces seules observations, deux modèles de comportement de l'appareil de Golgi ont été développés par les biologistes. Dans le modèle vésiculaire, la pile de saccules est fixe et les protéines sont transportées de saccule en saccule par l'intermédiaire des vésicules. Dans le modèle de la maturation, la pile suit un mouvement antérograde – c'est-à-dire du noyau vers la périphérie de la cellule – qui explique le transport des protéines. Dans ce cas, les vésicules seraient utilisées pour rapatrier des molécules – en fait des enzymes – selon un mouvement rétrograde. D'autres modèles de comportement plus récents font l'hypothèse d'un appareil de Golgi dans lequel les saccules seraient connectés par des tubes le long desquels sont transportées les protéines, c'est le cas du modèle du flux continu de membranes. Dans ce dernier modèle, les biologistes supposent de plus que les saccules sont parsemés de perforations. On remarque ainsi que ce sont principalement des caractéristiques topologiques et dynamiques qui permettent de distinguer ces trois modèles. Ainsi, afin d'aider les biologistes à discriminer ces différents modèles – c'est-à-dire à déterminer celui qui semble le plus proche des observations – nous avons cherché à les étudier, notamment par simulation, en nous focalisant sur ce qui caractérise leur topologie, ainsi que sur d'autres critères simples tels que le rapport de volume sur surface des différents compartiments de

l'appareil de Golgi.

Dans l'objectif de discriminer les trois modèles de l'appareil de Golgi, nous avons procédé à une étude statique de leur topologie. Pour cela, nous avons construit deux représentations tridimensionnelles de l'appareil de Golgi : une première pour les modèles continus puis une seconde pour le modèle du flux continu de membranes. Dans ces deux représentations – que nous avons construites à l'aide de cartes généralisées de dimension 3 – nous avons cherché à nous focaliser sur les aspects topologiques en abstrayant au maximum la géométrie. Notons que ces différentes représentations ont été validées par des biologistes spécialistes du Golgi. De plus, la construction du modèle continu leur a permis de visualiser pour la première fois cette hypothèse récente dans un espace en trois dimensions. Dans un second temps, nous avons effectué des coupes bidimensionnelles des modèles afin de comparer nos constructions aux observations faites par microscopie électronique sur de fines tranches de cellules. Il apparaît alors immédiatement que ce qui est usuellement interprété par les biologistes comme de petites vésicules peut également être interprété comme de fines sections de tubes. De plus, seule la coupe du modèle du flux continu de membranes permet d'expliquer certains artefacts parfois observés sur les images. En effet, la coupe des tubes du modèle continu permet d'expliquer la variété des formes des vésicules, tandis que la coupe des saccules perforés permet d'expliquer la discontinuité des saccules parfois observée sur les images. Ainsi, cette étude topologique statique nous a permis de crédibiliser le modèle du flux continu de membranes face aux modèles discontinus.

Nous avons également procédé à une étude de la dynamique des trois modèles de l'appareil de Golgi. Pour cela, nous avons effectué quelques simulations dans lesquels nous nous sommes focalisés sur un petit nombre de compartiments, auxquels nous avons attaché un volume et une surface. Lors des simulations, nous avons notamment étudié l'évolution du rapport de volume sur surface du saccule situé dans la région *trans* de l'appareil de Golgi. Ce saccule est le plus proche de la paroi de la cellule, il est le lieu depuis lequel sont excrétés les grains de sécrétion – que l'on peut voir comme de grosses vésicules – dont le rôle est de transporter les vésicules depuis la région *trans* vers le milieu extra-cellulaire. Dans le cas du modèle vésiculaire, la simulation montre une croissance non réaliste de la surface du saccule *trans*. Cette croissance s'explique par la différence de rapport de volume sur surface des vésicules qui entrent dans le saccule *trans* par rapport aux grains de sécrétion excrétés. Par calcul, nous avons montré que le rapport de volume sur surface des grains est plus élevé que celui des vésicules. Intuitivement, les grains de sécrétion présentent donc un déficit de membrane par rapport aux vésicules et ainsi pour chaque grain créé, une certaine quantité de membrane est stockée dans le saccule *trans*. À l'aide d'arguments analogues, nous avons décrédibilisé le modèle de la maturation. Dans ce dernier, les différences de rapports de volume sur surface entre les vésicules et les grains de sécrétion oblige l'appareil de Golgi à transporter une grande quantité de vésicules dans le sens rétrograde. Ces nombreux transports, dirigés vers le noyau de la cellule, permettent de rétablir l'équilibre entre la surface et le volume dans la région *trans* de l'appareil de Golgi. Cependant, d'après les biologistes, il résulte de ces nombreux allers et retours une inefficacité apparente du modèle de la maturation. Dans le modèle vésiculaire comme dans le modèle de la maturation, nous avons montré que l'ajout de tubes – initialement absents dans ces modèles – permet d'éviter l'accumulation de membrane dans la région *trans*. Ce faisant, nous avons à la fois montré que les deux modèles discontinus ne corroborent pas, en l'état, les observations effectuées par les biologistes, et avons démontré l'importance des tubes, déjà présents dans le modèle du flux continu de membranes.

Les différents travaux que nous avons menés autour de l'application à la biologie ont été effectués dans le cadre du groupe de travail GolgiTop. Dans ce groupe, nous avons travaillé de concert avec des biologistes spécialistes de l'appareil de Golgi. Nous avons ainsi pu mettre en pratique notre méthodologie de discrimination dans le cas de modèles simples basés sur une étude du volume et de la surface des différents compartiments de l'appareil de Golgi. Nous avons ainsi pu montrer, à la fois statiquement et dynamiquement, que le modèle du flux continu de membranes est crédible face aux modèles plus anciens basés sur l'hypothèse d'un appareil de Golgi discontinu. Enfin, nous avons également pu pro-

poser un premier prototype de simulateur basé sur la topologie. D'après nos collègues biologistes, ces premiers résultats – où l'étude « inédite » de la topologie a joué un rôle essentiel – sont encourageants et nous incitent à poursuivre cette collaboration afin de proposer à terme un outillage complet d'aide à la discrimination de modèles biologiques compartimentés.

Perspectives

Les perspectives que nous mentionnons ici ont été évoquées, pour la plupart, dans les chapitres 4 et 7. À court terme, les deux principales perspectives de nos travaux sont la manipulation de plusieurs variables orbites au sein d'une règle avec renommages ainsi que le traitement des plongements. Ces deux extensions nous permettront d'étendre sensiblement le pouvoir d'expression de nos règles de transformation de cartes généralisées. L'ajout de données de plongements à nos règles nous permettra en outre d'écrire de nouvelles règles dont l'application est contrôlée par des pré-conditions qui portent sur les plongements. Enfin, la manipulation de plusieurs variables ainsi que des plongements nous conduira à étendre les constructions utilisées lors de l'instanciation ainsi que les critères syntaxiques qui assurent la préservation des contraintes de cohérence des cartes généralisées plongées.

À moyen terme, nous devons implanter les règles de transformation munies de variables orbites et de plongements dans notre simulateur de modèles biologiques. De cette manière, nous pourrions simuler des structures topologiques où la structure des différentes cellules topologiques n'est pas fixée. Par exemple, une même règle de transformation de face pourra être appliquée quelle que soit la structure de la face, par exemple carrée ou triangulaire. Cette perspective implique notamment d'implanter les différentes constructions qui instancient les règles munies de variables orbites. Enfin, pour améliorer le réalisme des simulations, nous devons collaborer avec nos collègues biologistes afin d'étudier les stratégies d'application des règles qui semblent les plus adaptées à notre cas d'étude. Ceci nous conduira à traiter l'aspect temporel des simulations. En effet, bien que notre simulateur soit basé sur une boucle d'application de règles – et donc sur une discrétisation du temps – nous pourrions attribuer des fréquences ainsi que des durées d'application à nos différentes règles.

Notons que la poursuite de notre collaboration avec les biologistes au sein de GolgiTop sera essentielle afin d'affiner la méthodologie de discrimination que nous avons fait mûrir au travers de ce manuscrit. En particulier, les études expérimentales déjà mises en œuvre par les biologistes – notamment afin d'alimenter nos simulations avec des paramètres numériques réalistes – devront être poursuivies afin d'entreprendre une simulation du modèle du flux continu de membrane, plus complexe du point de vue topologique. En effet, pour ce modèle, une étude de l'évolution des rapports de volume sur surface n'est pas suffisante pour discriminer, puisque la connexion des saccules entre eux nous permet de considérer arbitrairement n'importe quel flux de matière dans le sens antérograde comme rétrograde. Ainsi, afin de rendre notre méthodologie de discrimination plus opérationnelle et de proposer un outillage complet pour l'aide à la discrimination, nous devons étudier d'autres critères de discrimination tels que des critères biochimiques ou encore énergétiques.

Enfin, à plus long terme, il serait intéressant d'étudier dans quelle mesure notre langage de transformation de cartes généralisées pourrait être utilisé pour d'autres modèles topologiques. En effet, la plupart des modèles topologiques peuvent être vus – à la manière des cartes généralisées – comme des cas particuliers de graphes. Par ailleurs, nous avons déjà étudié les analogies qui existent entre ces différents modèles, dans le cadre de la conception d'un générateur automatique de noyaux de modeleurs géométriques [Poudret 05]. Suite à cette première étude, nous envisageons d'utiliser nos règles de transformation de structures topologiques dans le cadre du développement d'un modeleur générique. Le développement de ce modeleur « générique par deux fois » – générique en dimension mais également générique en ce sens qu'il ne présuppose pas que l'on utilise un modèle topologique particulier – fait actuellement l'objet d'un travail de thèse au laboratoire XLIM-SIC de l'Université de Poitiers [Bellet 09]. L'idée à

la base de ce modeleur générique est proche du simulateur que nous proposons dans ce manuscrit. En effet, on propose de voir les opérations topologiques et géométriques du modeleur générique comme des règles de transformation de graphes. Le principal intérêt de cette approche est de réduire le modeleur à un moteur d'application de règles dont seuls les critères syntaxiques définis sur les règles, spécialisent le modeleur générique pour un modèle topologique donné. Remarquons qu'afin de rendre ce modeleur plus opérationnel, une grande importance devra être accordée à l'interface au travers de laquelle l'utilisateur entrera ses règles de transformations.

RAPPELS ET NOTATIONS

A.1 Les mots

Dans ce manuscrit, nous manipulons des mots construits sur un alphabet (dans notre cas un ensemble d'étiquettes notées α_0 à α_n qui représentent les relations de voisinage dans les structures topologiques). Nous rappelons ici quelques notions et notations classiques reprises de [Autebert 94].

Soit Σ un ensemble fini d'étiquettes. Un *mot* sur Σ est une suite finie d'étiquettes de Σ . Si $[p]$ désigne la suite d'entiers $(1, 2, \dots, p)$, un mot u est une application de $[p]$ dans Σ notée $u : [p] \rightarrow \Sigma$. p sera appelé la *longueur* du mot u , également notée $|u|$. Par conséquent, $u(i)$ désigne la i -ème étiquette de u pour $i \in [1, p]$.

Dans l'ensemble de tous les mots sur un alphabet Σ , on définit une opération, dite de *concaténation*, qui est une loi de composition interne, de la façon suivante :

soit u et v deux mots, la concaténation de u et v (dans cet ordre) est le mot noté uv de longueur $|u| + |v|$ défini par :

- $uv(i) = u(i)$ si $1 \leq i \leq |u|$
- $uv(p + i) = v(i)$ si $1 \leq i \leq |v|$

On vérifie très aisément que c'est une loi associative.

Par convention, on définit le mot de longueur nulle, *i.e.* l'application de \emptyset dans X , appelé le mot vide et noté ϵ , qui est l'élément neutre pour la concaténation.

L'ensemble de tous les mots sur l'alphabet X possède donc une structure de monoïde. Ce monoïde est noté X^* .

Le terme *morphisme* désignera toujours des homomorphismes de monoïdes. Si φ est un morphisme de X^* dans Y^* , noté $\varphi : X^* \rightarrow Y^*$, il est entièrement défini par l'image des lettres de X .

En effet, φ étant un morphisme, $\varphi(\epsilon) = \epsilon$ et $\forall u \in X^*, \forall x \in X : \varphi(ux) = \varphi(u)\varphi(x)$.

Dans notre cas, les morphismes $\varphi : X^* \rightarrow Y^*$ sont dits *strictement alphabétique*, c'est-à-dire $\forall x \in X : \varphi(x) \in Y$.

Nous dirons qu'un mot u de X^* a toutes ses lettres différentes, si pour toute lettre $l \in X$, il existe au plus une occurrence $i \in [1, |u|]$ tel que $u(i) = l$.

Nous noterons Σ_u , l'ensemble $\{u(i) \mid i \in [1, |u|]\}$ des lettres de u .

A.2 Théorie des graphes

Dans ce manuscrit, nous définissons le modèle topologique des cartes généralisées comme étant un cas particulier de graphes. Nous rappelons donc quelques notions classiques de la théorie des graphes ainsi que les notations correspondantes. Remarquons que nous présentons les graphes tels qu'il sont vus classiquement en théorie des catégories [Ehrig 06, Habel 02]. En particulier, nous introduisons la notion de morphisme de graphes.

A.2.1 Graphes

Définition 54 (Graphe). *Un graphe $G = (V, E, s, t)$ est¹ défini par :*

- un ensemble V de sommets ;
- un ensemble E d'arcs ;
- deux fonctions source $s : E \rightarrow V$ et cible $t : E \rightarrow V$ qui lient les arcs aux sommets. Soit $e \in E$, on dit que $s(e)$ et $t(e)$ sont respectivement le sommet source et le sommet cible de e . On dit que e est un arc incident à $s(e)$. Enfin, nous appelons boucle les arcs $e \in E$ tels que $s(e) = t(e)$.

Définition 55 (Graphe partiellement étiqueté). *Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp . Un graphe $G = (V, E, s, t, l_V, l_E)$ partiellement étiqueté sur (Σ_V, Σ_E) est défini de la manière suivante :*

- (V, E, s, t) est un graphe ;
- $l_V : V \rightarrow \Sigma_V \cup \{\perp\}$ est la fonction d'étiquetage des sommets. Soit v un sommet de V , si $l_V(v) = \perp$ on dit que v n'est pas étiqueté, sinon on dit que v est étiqueté par $l_V(v)$;
- $l_E : E \rightarrow \Sigma_E \cup \{\perp\}$ est la fonction d'étiquetage des arcs. Soit e un arc de E , si $l_E(e) = \perp$ on dit que e n'est pas étiqueté, sinon on dit que e est étiquetée par $l_E(e)$.

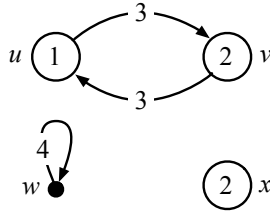
Exemple. Soient $\Sigma_V = \Sigma_E = \mathbb{N}$ et $G_1 = (V_{G_1}, E_{G_1}, s_{G_1}, t_{G_1}, l_{V_{G_1}}, l_{E_{G_1}})$ le graphe étiqueté sur (Σ_V, Σ_E) défini par :

- $V_{G_1} = \{u, v, w, x\}$;
- $E_{G_1} = \{(u, 3, v), (v, 3, u), (w, 4, w)\}$;
- $s_{G_1}((u, 3, v)) = u$, $s_{G_1}((v, 3, u)) = v$ et $s_{G_1}((w, 4, w)) = w$;
- $t_{G_1}((u, 3, v)) = v$, $t_{G_1}((v, 3, u)) = u$ et $t_{G_1}((w, 4, w)) = w$;
- $l_{V_{G_1}}(u) = 1$, $l_{V_{G_1}}(v) = 2$, $l_{V_{G_1}}(w) = \perp$ et $l_{V_{G_1}}(x) = 2$;
- $l_{E_{G_1}}((u, 3, v)) = 3$, $l_{E_{G_1}}((v, 3, u)) = 3$ et $l_{E_{G_1}}((w, 4, w)) = 4$.

Dans ce manuscrit, les arcs étiquetés sont souvent nommés par un triplet de $V \times \Sigma_E \times V$. Dans ce cas, on définit canoniquement s_G , t_G et l_{E_G} de la manière suivante : pour tout arc (x, l, y) de E , $s_G((x, l, y)) = x$, $t_G((x, l, y)) = y$ et $l_{E_G}((x, l, y)) = l$. Ainsi, dans G_1 , l'arc $(u, 3, v)$ est étiquetée par 3 et connecte son sommet source u à son sommet cible v . Remarquons que cette notation ne permet pas de dénoter deux arcs distincts qui portent la même étiquette et qui connectent le même sommet source au même sommet cible.

Une représentation graphique du graphe G_1 est proposée en Fig. A.1. Nous représentons les sommets non étiquetés par des points noirs, c'est le cas du sommet w de G_1 . Les sommets étiquetés sont représentés par des cercles décorés par leur étiquette. Par exemple, le sommet u – étiqueté par 1 – est représenté par un cercle portant une étiquette 1. Les arcs sont représentés par des flèches orientées depuis le sommet source vers le sommet cible. Les arcs étiquetés sont décorés par une étiquette. Par exemple, l'arc $(u, 3, v)$ est représenté par une flèche orientée depuis u vers v et décorée par l'étiquette 3.

1. Les notations V , E , s et t sont issues, respectivement, des mots anglais « vertice », « edge », « source » et « target ».

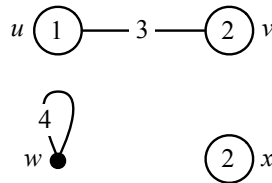
FIGURE A.1 – G_1

Définition 56 (Graphe non orienté). Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp et soit G un graphe étiqueté sur (Σ_V, Σ_E) .

Le graphe G est un graphe non orienté si et seulement si pour tout arc e de G de sommet source $s(e)$, de sommet cible $t(e)$ et d'étiquette $l_E(e)$, il existe un arc inverse e' dans G , c'est-à-dire de sommet source $s(e') = t(e)$, de sommet cible $t(e') = s(e)$ et d'étiquette $l_E(e') = l_E(e)$.

On appelle arête de G l'ensemble $\{e, e'\}$ constitué de deux tels arcs. On dit que l'arête $\{e, e'\}$ connecte les sommets $s(e)$ et $t(e)$, et est incidente aux sommets $s(e)$ et $t(e)$. De plus, si $l_E(e) = \perp$ on dit que $\{e, e'\}$ n'est pas étiquetée, sinon on dit que $\{e, e'\}$ est étiquetée par $l_E(e)$. Enfin, nous appelons boucles les arêtes de la forme $\{e, e\}$, c'est-à-dire telles que $s(e) = t(e)$.

Exemple. Remarquons que le graphe G_1 de la Fig. A.1 est un graphe non orienté. Notamment, les arcs $(u, 3, v)$ et $(v, 3, u)$ de G_1 sont deux arcs d'orientation inverse. Ils forment donc une arête qui connecte les sommets u et v et qui est étiquetée par 3. Graphiquement, nous représentons les arêtes par des traits – sans flèche – si besoin décorés par une étiquette. Nous pouvons alors représenter le graphe G_1 de la manière suivante.



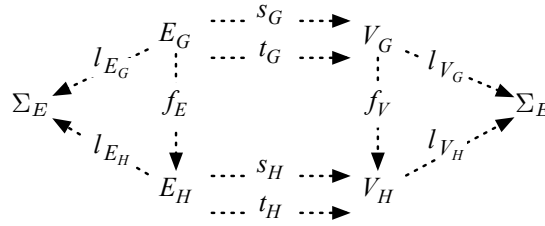
A.2.2 Morphismes de graphes

Définition 57 (Morphisme de graphes partiellement étiquetés). Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp et soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ et $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$ deux graphes partiellement étiquetés sur (Σ_V, Σ_E) .

Un morphisme de graphes partiellement étiquetés $f : G \rightarrow H$ est la donnée de deux fonctions $f_V : V_G \rightarrow V_H$ et $f_E : E_G \rightarrow E_H$ qui préservent la structure et l'étiquetage, c'est-à-dire les fonctions sources et cibles de connexion des arcs aux sommets et les fonctions d'étiquetage des sommets et des arcs. Plus formellement :

- $f_V \circ s_G = s_H \circ f_E$;
- $f_V \circ t_G = t_H \circ f_E$;
- pour tout sommet $v \in V_G$, si $l_{V_G}(v) \neq \perp$ alors $l_{V_H}(f_V(v)) = l_{V_G}(v)$;
- pour tout arc $e \in E_G$ de G , si $l_{E_G}(e) \neq \perp$ alors $l_{E_H}(f_E(e)) = l_{E_G}(e)$.

Sur la figure suivante, les fonctions sont représentées par des flèches en pointillés :



Graphiquement, on représente un morphisme de graphes étiquetés $f : G \rightarrow H$ de la manière suivante :

$$G \xrightarrow{f} H$$

Lorsque le nom du morphisme de graphes étiqueté importe peu, nous utilisons la notation simplifiée suivante :

$$G \longrightarrow H$$

Proposition 58. Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp . Soient G, H et I trois graphes étiquetés sur (Σ_V, Σ_E) et $f : G \rightarrow H$ et $g : H \rightarrow I$ deux morphismes de graphes partiellement étiquetés.

La composition $(g \circ f) = ((g_V \circ f_V), (g_E \circ f_E)) : G \rightarrow I$ est un morphisme de graphes étiquetés.

Définition 59 (Morphisme d'inclusion de graphes partiellement étiquetés). Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp et soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ et $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$ deux graphes partiellement étiquetés sur (Σ_V, Σ_E) .

Le morphisme d'inclusion de G dans H , lorsqu'il existe, est l'unique morphisme de graphes partiellement étiquetés $f : G \rightarrow H$ tel que :

- pour tout sommet v de V_G , $f_V(v) = v$;
- pour tout arc e de E_G , $f_E(e) = e$;

dans ce cas, on dit que G est inclus dans H – ou encore que G est un sous-graphe de H – et on note $f : G \hookrightarrow H$.

Lorsque f est surjectif, il est le morphisme d'identité de G et H . Dans ce cas, on dit que G et H sont égaux et on note $G = H$.

Exemple. Soient $\Sigma_V = \Sigma_E = \mathbb{N}$ et $G_2 = (V_{G_2}, E_{G_2}, s_{G_2}, t_{G_2}, l_{V_{G_2}}, l_{E_{G_2}})$ le graphe étiqueté sur (Σ_V, Σ_E) illustré en Fig. A.2.

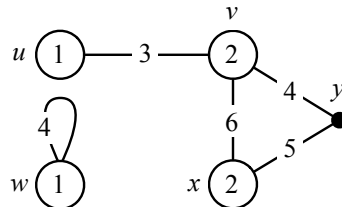
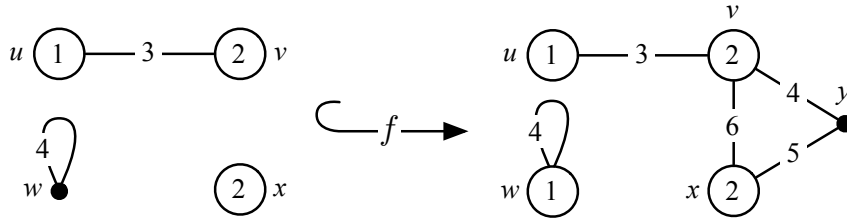


FIGURE A.2 – G_2

Le graphe étiqueté G_1 de la Fig. A.1 page 209 est tel qu'il existe un morphisme d'inclusion de graphes étiquetés $f = (f_V, f_E) : G_1 \hookrightarrow G_2$ défini par :

- les images des sommets u, v, w et x de G_1 par f_V sont respectivement les sommets u, v, w et x de G_2 ;
- les images de arcs $(u, 3, v)$ et $(v, 3, u)$ de G_1 par f_E sont respectivement les arcs $(u, 3, v)$ et $(v, 3, u)$ de G_2 .

Nous représentons graphiquement le morphisme d'inclusion de graphes étiquetés f de la manière suivante :



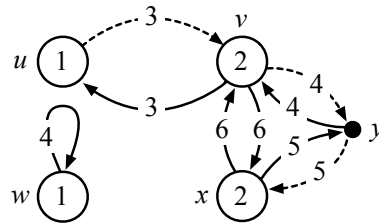
Les fonctions f_V et f_E sont déterminées par les noms des sommets. Remarquons que f est bien un morphisme de graphes partiellement étiquetés. Notamment, les sommets et arcs étiquetés de G_1 possèdent les mêmes étiquettes dans G_2 . Par exemple, le sommet u étiqueté par 1 dans G_1 est également étiqueté par 1 dans G_2 . Enfin, remarquons que le morphisme f « ajoute des étiquettes », ce qui est permis par la définition 57 des morphismes de graphes étiquetés. En effet, le sommet w , qui n'est pas étiqueté dans G_1 , est étiqueté par 1 dans G_2 .

A.2.3 Notions complémentaires

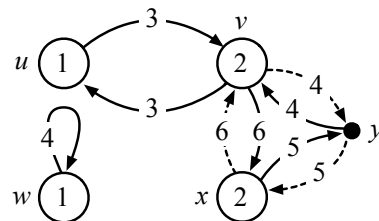
Définition 60 (chemin). Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp et $G = (V, E, s, t, l_V, l_E)$ un graphe partiellement étiqueté sur (Σ_V, Σ_E) .

Un chemin dans G est une séquence d'arcs de G $e_1 e_2 \dots e_k$ telle que pour tout entier i tel que $1 \leq i < k$, $t(e_i) = s(e_{i+1})$. On dit que le chemin $e_1 e_2 \dots e_k$ a pour source le sommet $s(e_1)$, pour cible le sommet $t(e_k)$ et est étiqueté par le mot $l_E(e_1) l_E(e_2) \dots l_E(e_k)$. De plus, si $s(e_1) = t(e_k)$, un tel chemin est appelé cycle.

Exemple. Dans le graphe G_2 de la Fig. A.2 page 210, il existe un chemin $(u, 3, v)(v, 4, y)(y, 5, x)$ qui connecte le sommet u au sommet x . Ce chemin est illustré en pointillés sur la figure suivante :



De même, dans le graphe G_2 , il existe un cycle $(v, 4, y)(y, 5, x)(x, 6, v)$. Ce cycle est illustré en pointillés sur la figure suivante :

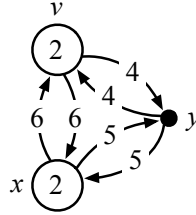


Définition 61 (sous-graphe généré). Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp , $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ un graphe étiqueté sur (Σ_V, Σ_E) , v un sommet de G et $\Sigma'_E \subset \Sigma_E$ un ensemble d'étiquettes d'arcs.

Le sous-graphe de G généré par v et Σ'_E est le plus grand² graphe $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$ étiqueté sur (Σ_V, Σ'_E) inclus dans G tel que :

- v est un sommet de H ;
- pour tout sommet u de H , il existe un chemin qui connecte v à u dans H .

Exemple. Le graphe suivant est le sous-graphe de H généré par v et $\{4, 5, 6\}$:



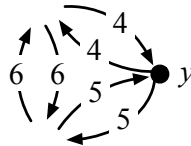
Définition 62 (soustraction de deux graphes partiellement étiquetés). Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp et soient $G = (V_G, E_G, s_G, t_G, l_{V_G}, l_{E_G})$ et $H = (V_H, E_H, s_H, t_H, l_{V_H}, l_{E_H})$ deux graphes partiellement étiquetés sur (Σ_V, Σ_E) .

La soustraction de H à G , notée $G \setminus H$, est un sexuplet $(V_{G \setminus H}, E_{G \setminus H}, s_{G \setminus H}, t_{G \setminus H}, l_{V_{G \setminus H}}, l_{E_{G \setminus H}})$ défini par :

- $V_{G \setminus H} = V_G \setminus V_H$;
- $E_{G \setminus H} = E_G \setminus E_H$;
- $s_{G \setminus H} : E_{G \setminus H} \rightarrow V_{G \setminus H}$. Soit un arc $e \in G \setminus H$, $s_{G \setminus H}(e)$ est défini si et seulement si $s_G(e) \in V_{G \setminus H}$, et dans ce cas $s_{G \setminus H}(e) = s_G(e)$;
- $t_{G \setminus H} : E_{G \setminus H} \rightarrow V_{G \setminus H}$. Soit un arc $e \in G \setminus H$, $t_{G \setminus H}(e)$ est défini si et seulement si $t_G(e) \in V_{G \setminus H}$, et dans ce cas $t_{G \setminus H}(e) = t_G(e)$;
- $l_{V_{G \setminus H}} : V_{G \setminus H} \rightarrow \Sigma_V$ est tel que pour tout sommet $v \in V_{G \setminus H}$, $l_{V_{G \setminus H}}(v) = l_{V_G}(v)$;
- $l_{E_{G \setminus H}} : E_{G \setminus H} \rightarrow \Sigma_E$ est tel que pour tout arc $e \in E_{G \setminus H}$, $l_{E_{G \setminus H}}(e) = l_{E_G}(e)$.

En général, la structure $G \setminus H$ n'est pas un graphe étiqueté car les fonctions $s_{G \setminus H}$ et $t_{G \setminus H}$ ne sont pas totalement définies sur $E_{G \setminus H}$.

Exemple. La structure suivante est la soustraction de G_1 (voir Fig. A.1 page 209) à G_2 (voir Fig. A.2 page 210), $G_1 \setminus G_2$:



A.3 Théorie des catégories

Dans ce manuscrit, nous utilisons les diagrammes commutatifs, les sommes amalgamées ainsi que les produits fibrés de la théorie des catégories [Ehrig 06, Habel 02]. Ces différentes notions nous permettent notamment de coller des graphes entre eux et d'effectuer des produits de graphes. Nous rappelons ici quelques notions classiques de la théorie des catégories et introduisons notamment la catégorie

2. Au sens de l'inclusion.

des graphes partiellement étiquetés.

A.3.1 Catégories

Définition 63 (catégorie). Une catégorie $\mathbb{C} = (Ob, Mor, \circ, id)$ est définie par

- une classe Ob d'objets ;
- pour toute paire d'objets $A, B \in Ob$, un ensemble $Mor(A, B)$ de morphismes. Un élément f de $Mor(A, B)$ est noté $f : A \rightarrow B$;
- pour tous objets $A, B, C \in Ob$, une opération de composition $\circ_{(A,B,C)} : Mor(B, C) \times Mor(A, B) \rightarrow Mor(A, C)$;
- pour tout objet $A \in Ob$, un morphisme d'identité $id_A \in Mor(A, A)$;

tel que les propriétés suivantes sont satisfaites :

Associativité : pour tous objets $A, B, C, D \in Ob$ et pour tout triplet de morphismes $f : A \rightarrow B$, $g : B \rightarrow C$ et $h : C \rightarrow D$, la propriété suivante est vérifiée : $(h \circ g) \circ f = h \circ (g \circ f)$.

Identité : pour tous objets $A, B \in Ob$ et pour tout morphisme $f : A \rightarrow B$, on a : $f \circ id_A = f$ et $id_B \circ f = f$.

Définition 64 (catégorie des graphes partiellement étiquetés). Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp . La catégorie $\mathbb{G}e(\Sigma_V, \Sigma_E) = (Ob_{\mathbb{G}e}, Mor_{\mathbb{G}e}, \circ_{\mathbb{G}e}, id_{\mathbb{G}e})$ des graphes partiellement étiquetés sur (Σ_V, Σ_E) est définie par :

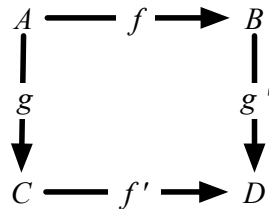
- la classe $Ob_{\mathbb{G}e}$ des graphes étiquetés sur (Σ_V, Σ_E) (voir définition 55 page 208) ;
- l'ensemble $Mor_{\mathbb{G}e}$ des morphismes de graphes étiquetés sur (Σ_V, Σ_E) (voir définition 57 page 209) ;
- la composition $\circ_{\mathbb{G}e}$ des morphismes de graphes étiquetés sur (Σ_V, Σ_E) (voir proposition 58 page 210) ;
- l'ensemble $id_{\mathbb{G}e}$ des morphismes d'identité de graphes étiquetés sur (Σ_V, Σ_E) (voir définition 59 page 210 des morphismes d'inclusion de graphes étiquetés).

A.3.2 Diagrammes commutatifs

Définition 65 (diagramme commutatif). Soient $\mathbb{C} = (Ob, Mor, \circ, id)$ une catégorie, A, B et C trois objets de Ob et deux morphismes $f : A \rightarrow B$ et $g : A \rightarrow C$ de Mor . Un diagramme commutatif (D, f', g') sur f et g est défini par :

- un objet D de Ob ;
- deux morphismes $f' : C \rightarrow D$ et $g' : B \rightarrow D$ tels que $f' \circ g = g' \circ f$.

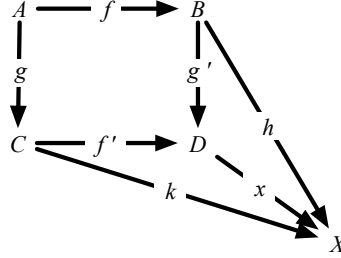
Graphiquement, un diagramme commutatif (D, f', g') sur f et g est représenté de la manière suivante :



A.3.3 Sommes amalgamées

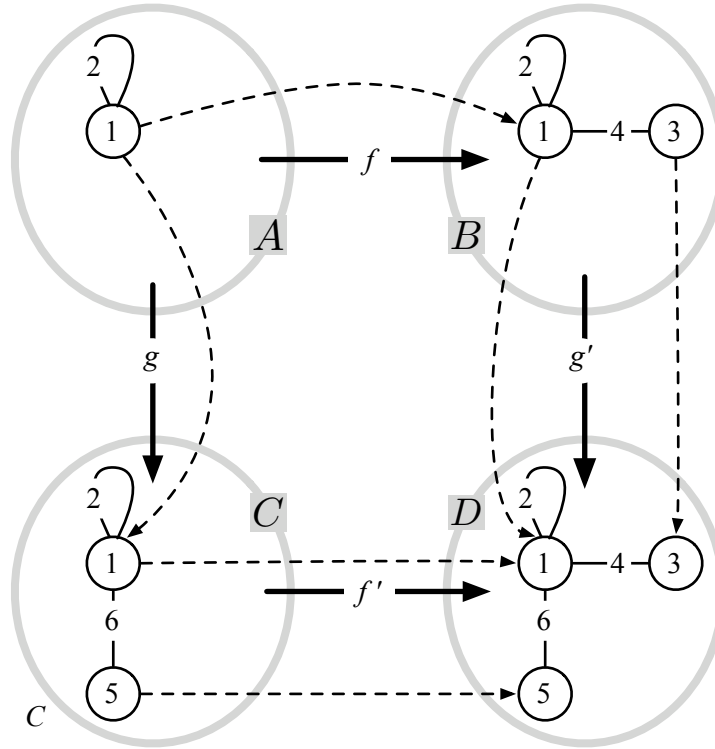
Définition 66 (somme amalgamée). Soient $\mathbb{C} = (Ob, Mor, \circ, id)$ une catégorie, A, B et C trois objets de Ob et deux morphismes $f : A \rightarrow B$ et $g : A \rightarrow C$ de Mor . Un diagramme commutatif (D, f', g') sur

f et g est une somme amalgamée – ou pushout – lorsque la propriété universelle suivante est satisfaite : pour tout objet X et pour tout couple de morphismes $h : B \rightarrow X$ et $k : C \rightarrow X$ tels que $k \circ g = h \circ f$, il existe un unique morphisme $x : D \rightarrow X$ tel que $x \circ g' = h$ et $x \circ f' = k$:



On écrit $D = B +_A C$ et on dit que l'objet D est le résultat du collage de B et C le long de A .

Exemple. L'existence des sommes amalgamées dans la catégorie des graphes partiellement étiquetés est montrée dans [Habel 02]. Un exemple de somme amalgamée dans la catégorie des graphes partiellement étiquetés $\mathbb{G}e(\mathbb{N}, \mathbb{N})$ – concernant des graphes non orientés – est donné dans la figure ci-dessous :



Sur la figure, les morphismes $f = (f_V, f_E)$, $g = (g_V, g_E)$, $f' = (f'_V, f'_E)$ et $g' = (g'_V, g'_E)$ sont dénotés par les flèches en pointillés. Par exemple, l'image par f_V du sommet de A étiqueté par 1 est le sommet étiqueté par 1 dans B . On en déduit que l'image par f_E de la boucle étiquetée par 2 dans A est la boucle étiquetée par 2 dans B .

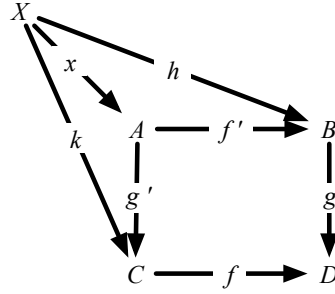
Dans ce manuscrit, nous utilisons les diagrammes commutatifs afin de coller des graphes entre eux le long d'un sous-graphe commun. Ainsi, sur la figure, le graphe D est le résultat du collage des graphes B et C le long de A .

A.3.4 Produits fibrés

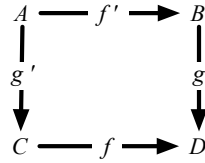
Définition 67 (Produit fibré). Soient $\mathbb{C} = (Ob, Mor, \circ, id)$ une catégorie, B, C et D trois objets de Ob et deux morphismes $f : C \rightarrow D$ et $g : B \rightarrow D$ de Mor . Un produit fibré – ou pullback – (A, f', g') sur f et g est défini par :

- un objet A de $Ob_{\mathbb{C}}$;
- deux morphismes $f' : A \rightarrow B$ et $g' : A \rightarrow C$ tels que $g \circ f' = f \circ g'$;

Le produit fibré (A, f', g') est tel que la propriété universelle suivante soit satisfaite : pour tout objet X et pour tout couple de morphismes $h : X \rightarrow B$ et $k : X \rightarrow C$ tels que $f \circ k = g \circ h$, il existe un unique morphisme $x : X \rightarrow A$ tel que $f' \circ x = h$ et $g' \circ x = k$:



Graphiquement, un produit fibré (A, f', g') est représenté de la manière suivante :

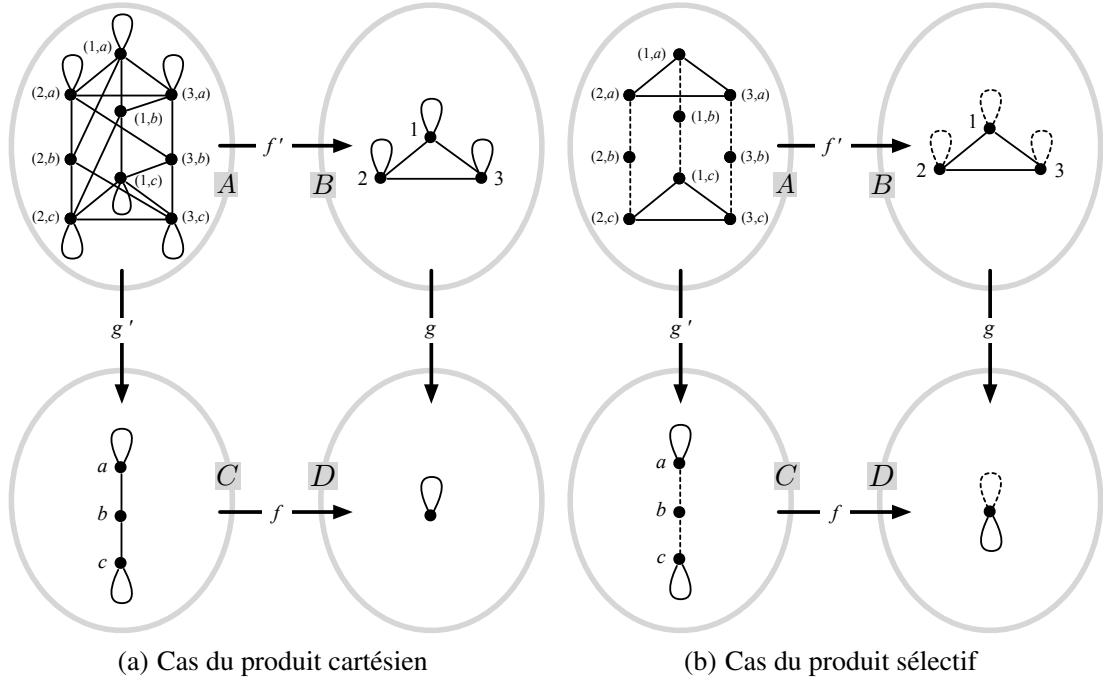


Exemple. L'existence des produits fibrés dans la catégorie des graphes partiellement étiquetés est montrée dans [Habel 02]. Deux exemples de produits fibrés dans la catégorie des graphes partiellement étiquetés $\mathbb{Ge}(\emptyset, \emptyset)$ – concernant des graphes non orientés – sont donnés en Fig. A.3. Dans ce manuscrit, nous utilisons les produits fibrés afin de calculer des produits de graphes. Intuitivement, avec la donnée de trois graphes B, C et D ainsi que de deux morphismes de graphes étiquetés $f : C \rightarrow D$ et $g : B \rightarrow D$, le produit fibré calcule le graphe A résultat du produit de B et C . La structure de A dépend de la projection des graphes B et C sur D , respectivement, par les morphismes g et f .

Dans l'exemple de la Fig. A.3(a), le graphe D est constitué d'un unique sommet muni d'une unique arête incidente. Le morphisme f est l'unique morphisme que l'on peut construire de C vers D . Il associe à chaque sommet de C l'unique sommet de D et à chaque arête de C l'unique arête de D . Le morphisme g est également l'unique morphisme que l'on peut construire de B vers D . Dans ce premier exemple, le graphe résultat A est isomorphe au *produit cartésien* des graphes B et C . Son ensemble de sommet (respectivement d'arêtes) est isomorphe au produit cartésien des sommets (respectivement des arêtes) de B et C .

Dans l'exemple de la Fig. A.3(b), on effectue un *produit sélectif* de B et C . Le graphe D est constitué d'un unique sommet muni d'exactly deux arêtes incidentes. Les morphismes f et g sont tels que les sommets de C et B sont projetés sur l'unique sommet de D et tel que les arêtes représentées en trait plein (respectivement en pointillés) sont projetées sur l'arête représentée en trait plein (respectivement en pointillés) dans D . Nous parlons de produit sélectif dans ce cas particulier où les arcs de B et C ne sont pas tous projetés sur un même arc de D .

La construction du produit fibré de deux graphes partiellement étiquetés est introduite dans la proposition suivante.

FIGURE A.3 – Exemples de produits fibrés dans la catégorie des graphes partiellement étiquetés $\mathcal{Ge}(\emptyset, \emptyset)$

Proposition 68. Soient Σ_V et Σ_E respectivement un ensemble d'étiquettes de sommets et un ensemble d'étiquettes d'arcs qui ne contiennent pas l'étiquette \perp . Plaçons nous dans la catégorie des graphes partiellement étiquetés $\mathcal{Ge}(\Sigma_V, \Sigma_E)$. Soient $B = (V_B, E_B, s_B, t_B, l_{V_B}, l_{E_B})$, $C = (V_C, E_C, s_C, t_C, l_{V_C}, l_{E_C})$ et D trois graphes et soient $f = (f_V, f_E) : B \rightarrow D$ et $g = (g_V, g_E) : C \rightarrow D$ deux morphismes.

Le produit fibré (A, f', g') sur f et g est construit comme suit :

- l'ensemble des sommets V_A de A est l'ensemble $\{(u, v) \in V_B \times V_C \mid f_V(u) = g_V(v)\}$;
- l'ensemble des arcs E_A de A est l'ensemble $\{(e, e') \in E_B \times E_C \mid f_E(e) = g_E(e')\}$;
- tout arc $(e, e') \in E_A$ de A a pour source le sommet $(s_B(e), s_C(e'))$ et pour cible $(t_B(e), t_C(e'))$;
- tout sommet $(u, v) \in V_A$ de A a pour étiquette $l_{V_B}(u)$ si $l_{V_B}(u) = l_{V_C}(v) \neq \perp$ et \perp sinon ;
- tout arc $(e, e') \in E_A$ de A a pour étiquette $l_{E_B}(e)$ si $l_{E_B}(e) = l_{E_C}(e') \neq \perp$ et \perp sinon ;
- $f' = (f'_V, f'_E)$ et $g' = (g'_V, g'_E)$ sont tels que pour tout sommet $(u, v) \in V_A$, $f'_V((u, v)) = v$ et $g'_V((u, v)) = u$ et pour tout arc $(e, e') \in E_A$, $f'_E((e, e')) = e'$ et $g'_E((e, e')) = e$.

Démonstration. La construction du produit fibré est faite dans [Habel 02] (voir la preuve du lemme 1). □

Exemple. Dans les exemples de produits fibrés proposés en Fig. A.3, conformément à la proposition précédente, nous avons nommé les sommets de A par des couples de sommets de $B \times C$.

Illustrons à présent la proposition. Dans l'exemple de la Fig. A.3(a), le sommet 1 de B et le sommet a de C sont projetés sur le même sommet dans D (en fait l'unique sommet de D). Par conséquent, il existe dans A un unique sommet issu de la projection de 1 et a dans D , nommé $(1, a)$ par convention. De même les arcs qui connectent les sommets 1 et 2 dans B et les sommets a et b dans C sont projetés sur le même arc dans D . Il existe donc dans A un arc qui connecte les sommets $(1, a)$ et $(2, b)$. Remarquons enfin que comme tous les sommets de B et C sont projetés sur le même sommet dans D , l'ensemble des sommets de A est donc le produit cartésien des sommets de B et C . Il en est de même pour les arcs de A .

Dans l'exemple de la Fig. A.3(b), dans le graphe A , aucun arc ne connecte les sommet $(1, a)$ et $(2, b)$. En effet, il existe bien dans B un arc qui connecte les sommets 1 et 2 et dans C un arc qui connecte les sommets a et b . Cependant, ces deux arcs sont projetés sur deux arcs distincts dans D .

PLONGEMENTS DU SIMULATEUR

B.1 Plongement des demi-faces

Dans une 3-G-carte, les orbites $\langle \alpha_0 \alpha_1 \rangle$ sont également appelées *demi-faces*. Dans notre simulateur, en plus des volumes, nous plongeons les demi-faces. Sur la Fig. B.1 sont illustrées trois demi-faces issues du modèle du transport. Les demi-faces F_1 et F_2 , munies respectivement des brins b_{16} à b_{23} et b_0 à b_7 appartiennent au compartiment A. La demi-face F_3 , qui comprend les brins b_8 à b_{15} , appartient au tube. Nous modélisons les membranes des compartiments biologiques par les demi-faces qui bordent les volumes. Par exemple, F_1 modélise une portion de la membrane du compartiment A. Comme pour les compartiments, à chaque portion de membrane est associé un type ainsi que des protéines membranaires. En ce sens, le plongement des demi-faces est analogue au plongement des volumes. Cependant, sur la Fig. B.1, on remarque que certaines demi-faces ne modélisent pas une membrane mais la connexion entre deux compartiments, nous les appelons *demi-faces virtuelles*. Sur la figure, les demi-faces F_2 et F_3 sont virtuelles. Sans elles, le compartiment A et le tube seraient réunis en un unique volume.

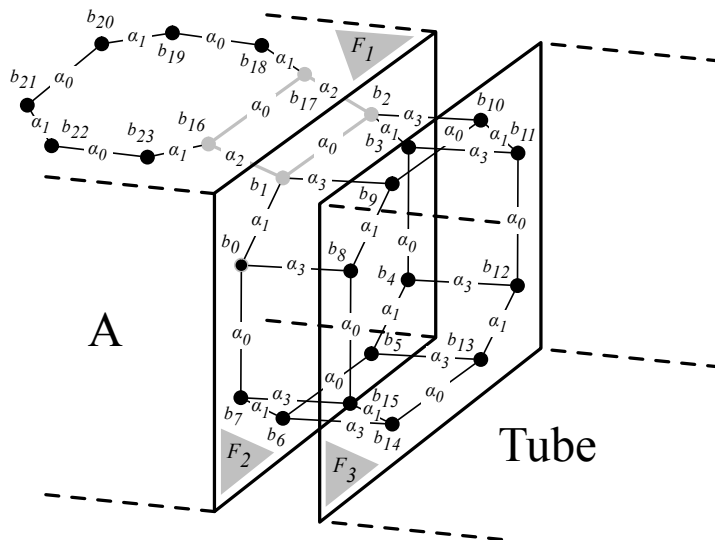


FIGURE B.1 – Trois demi-faces dans l'exemple du transport de protéines le long d'un tube

Les demi-faces virtuelles modélisent également la viscosité des compartiments. La viscosité est un paramètre important en biologie. En effet, les compartiments biologiques sont parfois vus – à tort – comme des poches remplies d’eau dans lesquelles baignent quelques molécules. Or, les compartiments sont habités par une population gigantesque d’espèces chimiques, augmentant ainsi la viscosité du milieu. De plus, à l’échelle de la cellule, même l’eau est vue comme un ensemble de molécules qui interagissent avec les autres espèces moléculaires. Une manière d’abstraire cette sur-population des compartiments biologiques est de considérer que le transport d’une molécule donnée – dans notre cas une protéine – est freiné par une viscosité ambiante. Afin de modéliser la viscosité, nous associons une *perméabilité* aux demi-faces virtuelles qui séparent les compartiments. Dans le modèle du transport, nous supposons que les protéines circulent à la même vitesse de A vers le tube et du tube vers A. Ainsi, nous associons la même perméabilité aux demi-faces F_2 et F_3 . Si nous souhaitons privilégier l’une des deux directions, il suffit de leur attribuer une perméabilité différente.

```

1 type t_plongement_demi_face =
2   Demi_Face_Défaut
3   | Demi_Face_Virtuelle of (float list) (* liste de perméabilités *)
4   | Demi_Face of t_plongement_demi_face_utilisateur

```

Code B.1 – Type Caml pour le plongement des demi-faces

Le type Caml du plongement des demi-faces, que nous nommons `t_plongement_demi_face`, est introduit dans le Code B.1. Les types biologiques `Demi_Face_Défaut` et `Demi_Face` sont respectivement analogues aux types biologiques `Volume_Défaut` et `Volume`. De plus, nous introduisons un type biologique `Demi_Face_Virtuelle` qui caractérise les demi-faces virtuelles. Le constructeur `Demi_Face_Virtuelle` prend en paramètre une liste de valeurs flottantes de type `float`. Chaque élément de la liste est une perméabilité associée à une espèce de protéine soluble.

```

1 type t_plongement_demi_face_utilisateur =
2   Demi_Face_A of (t_protéine list)
3   | Demi_Face_B of (t_protéine list)
4   | Demi_Face_Tube of (t_protéine list)

```

Code B.2 – Type Caml utilisateur pour le plongement des demi-faces dans le modèle du transport

Le type Caml utilisateur pour le plongement des demi-faces dans le modèle du transport est introduit dans le Code B.2. Il est analogue au type `t_plongement_volume_utilisateur`. Cette fois-ci, nous manipulons des listes de protéines membranaires.

Sur la Fig. B.2, nous reprenons les brins b_1 , b_2 , b_{16} et b_{17} de la Fig. B.1 et leur attribuons une donnée de plongement de volume et de demi-face. Notons que comme ces brins appartiennent tous aux compartiment A, ils présentent le même plongement de volume. Les brins b_1 et b_2 appartiennent à la demi-face virtuelle F_2 , ils portent le plongement de demi-face `Face(Face_Virtuelle(pf))` où `pf` est une valeur du type Caml `float list`. Les brins b_{16} et b_{17} appartiennent à la demi-face F_1 qui modélise une portion de la membrane du compartiment A. Ils portent tous les deux le plongement de demi-face `Face(Face_A(pro_3))` où `pro_3` est une valeur du type Caml `t_protéine list`.

B.2 Plongement des arêtes simples

Il existe un équivalent en dimension 1 de la notion de demi-face ; dans une 3-G-carte, le terme *arête simple* désigne deux brins connectés par α_0 . Par exemple, sur la Fig. B.1, les couples (b_{16}, b_{17}) et (b_1, b_2) sont respectivement une arête simple de F_1 et de F_2 . Le type Caml `t_plongement_arête_simple`

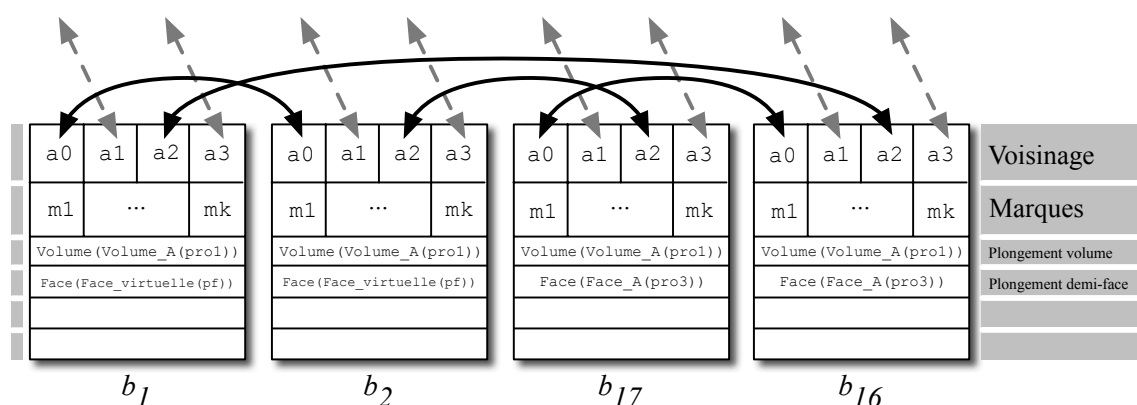


FIGURE B.2 – Exemple de plongement des demi-faces

```

1 type t_plongement_arête_simple =
2   Arête_Simple_Défaut
3   | Arête_Simple_Virtuelle of (float list) (* liste de perméabilités
      *)

```

Code B.3 – Type Caml générique pour le plongement des arêtes simples

introduit dans le Code B.3 est le type Caml des valeurs de plongement des arêtes simples. À l'image des demi-faces virtuelles qui modélisent la viscosité des compartiments, les *arêtes simples virtuelles* modélisent la viscosité des membranes. Dans l'exemple de la Fig. B.1, des protéines membranaires sont associées à la demi-face F_1 et sont notamment transportées sur la demi-face supérieur du tube (pour simplifier la Fig. B.1, nous ne la représentons pas explicitement). L'arête simple (b_1, b_2) contrôle le transport des protéines membranaires de F_1 vers la membrane du tube, elle est de type biologique *Arête_Simple_Virtuelle*. À l'image du constructeur *Demi_Face_Virtuelle* du Code B.1, *Arête_Simple_Virtuelle* prend une liste de perméabilités en paramètre. La Fig. B.3 illustre les données de plongement d'arêtes simples des brins b_1, b_2, b_{17} et b_{16} . Les arêtes simples non virtuelles sont de type biologique *Arête_Simple_Défaut*.

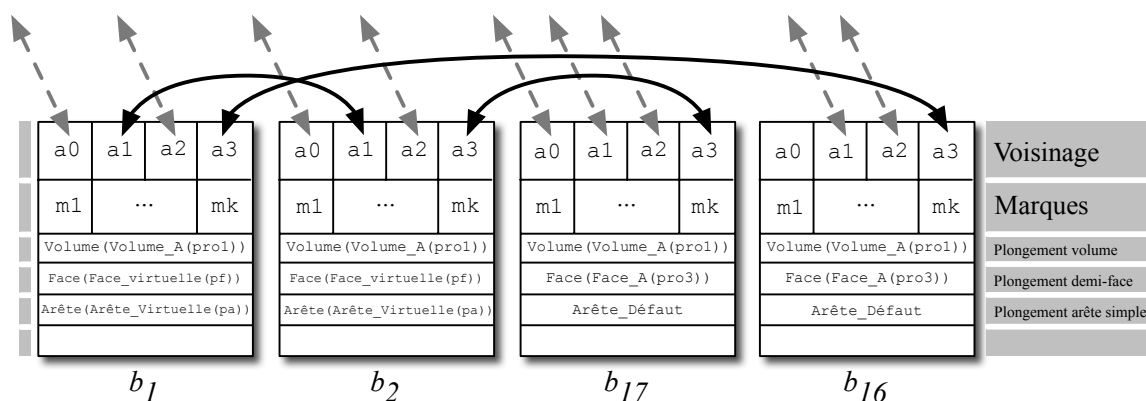


FIGURE B.3 – Exemple de plongement des arêtes simples

B.3 Plongement des 0-cellules

Dans notre simulateur nous utilisons un plongement polyédrique. Ainsi, un sommet géométrique 3D est associé à chaque 0-cellule. Le type Caml des données de plongement des 0-cellules est introduit dans le Code B.4. Sur la Fig. B.4 les brins b_1 , b_2 , b_{16} et b_{17} – issus de la Fig. B.1 – possèdent toutes leurs données de plongement. Les valeurs $s1$ et $s2$ sont de type `t_plongement_sommet`. Remarquons que les brins b_1 et b_{16} – qui appartiennent à la même 0-cellule – ont le même plongement de 0-cellule, à savoir $s1$. Les brins b_2 et b_{17} portent également la même donnée de plongement 0-cellule.

```
1 type t_plongement_sommet = float * float * float
```

Code B.4 – Type Caml générique pour le plongement des 0-cellules

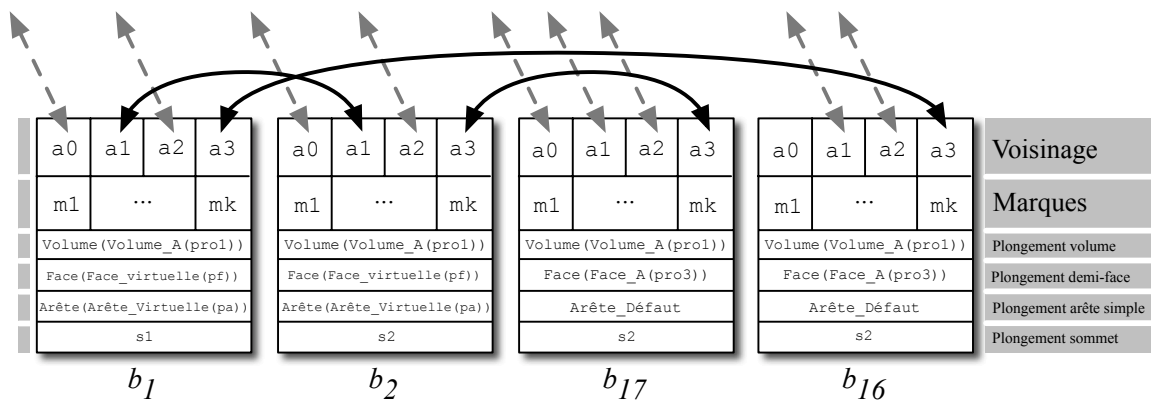


FIGURE B.4 – Illustration du plongement dans notre simulateur

B.4 Exemple du bourgeonnement des vésicules

Dans le modèle du bourgeonnement, nous manipulons un saccule et des vésicules. Nous utilisons alors deux types biologiques : `Volume_Saccule` et `Volume_Vesicule`, introduit dans le type Caml `t_plongement_volume_utilisateur` du Code B.5. À chacun de ces constructeurs est associée une liste qui modélise des protéines solubles.

```
1 type t_plongement_volume_utilisateur =
2   Volume_Saccule of (t_protéine list)
3   | Volume_Vesicule of (t_protéine list)
```

Code B.5 – Type Caml utilisateur pour le plongement des volumes du modèle du bourgeonnement

Dans ce même modèle, il n'y a pas de protéine membranaire. Il nous suffit de distinguer les portions de membranes selon qu'elles portent ou non un activateur. Les types biologiques des demi-faces sont introduits dans le type Caml `t_plongement_demi_face_utilisateur` du Code B.6. Nous supposons que les vésicules bourgeonnent sur les bords latéraux des saccules. Ainsi, nous utilisons le type biologique `Demi_Face_Bord` pour distinguer les demi-faces latérales des demi-faces supérieures et inférieures. Ces dernières sont de type biologique `Demi_Face_Défaut`. Nous devons à présent distinguer la demi-face qui, parmi les demi-faces latérales, porte l'activateur responsable du bourgeonnement des vésicules. Pour cela, nous utilisons le type `Demi_Face_Bord_Activateur`. Remarquons

qu'aucun de ces constructeurs n'a de paramètre.

```
1 type t_plongement_demi_face_utilisateur =  
2   Demi_Face_Bord  
3   | Demi_Face_Bord_Activateur
```

Code B.6 – Type Caml utilisateur pour le plongement des demi-faces du modèle du bourgeonnement

BIBLIOGRAPHIE

- [3ds] *3ds Max*. Logiciel commercial de modélisation, d'animation et de rendu 3D développé par Autodesk. www.autodesk.com.
- [Agoston 76] M. K. Agoston. Algebraic topology. a first course. Numeéro 32 in Monographs and textbooks in pure and applied mathematics. 1976.
- [Alberts 02] B. Alberts. Molecular biology of the cell, 4th ed. Garland Science, 2002.
- [Allaire 07] G. Allaire. Numerical analysis and optimization : An introduction to mathematical modelling and numerical simulation. Oxford University Press, 2007.
- [Andries 99] M. Andries, G. Engels, A. Habel, B. Hoffmann, H.-J. Kreowski, S. Kuske, D. Plump, A. Schurr & G. Taentzer. *Graph transformation for specification and programming*. Science of Computer Programming, vol. 34, pages 1–54, 1999.
- [Autebert 94] J.-M. Autebert. Théorie des langages et des automates. Masson, 1994.
- [Bauderon 01] Michel Bauderon & Jacquet Hélène. *Pullback as a Generic Graph Rewriting Mechanism*. Applied Categorical Structures, vol. 9, no. 1, pages 65–82, janvier 2001.
- [Baumgart 75] B. Baumgart. *A polyhedron representation for computer vision*. In AFIPS Nat. Conf. Proc., volume 44, pages 589–596, 1975.
- [Bechmann 07a] D. Bechmann & B. Péroche, éditeurs. Informatique graphique et rendu, volume 2 of *Traitement du Signal et de l'Image*. Hermès, 2007.
- [Bechmann 07b] D. Bechmann & B. Péroche, éditeurs. Informatique graphique, modélisation géométrique et animation, volume 1 of *Traitement du Signal et de l'Image*. Hermès, 2007.
- [Bellet 08] T. Bellet, A. Arnould, S. Charneau & L. Fuchs. *Modélisation nD à base d'algèbres géométriques*. In AFIG 2008, Toulouse, November 2008.
- [Bellet 09] T. Bellet, A. Arnould, Y. Bertrand & L. Fuchs. *Opérations topologiques nD à base de règles*. In GTMG 2009, Toulouse, Mars 2009.
- [Bertrand 92] Y. Bertrand. *Spécification algébrique et réalisation d'un modèleur interactif d'objets géométriques volumiques*. PhD thesis, Université Louis-Pasteur de Strasbourg, 1992.
- [Bertrand 98] Y. Bertrand. *Topofil : un modèleur interactif d'objets 3D à base topologique*. vol. 17, no. 4, pages 443–483, Avril 1998. T.S.I.
- [Ble] *Blender*. Logiciel libre de modélisation, d'animation et de rendu 3D. www.blender.org.
- [Bonfanti 98] L. Bonfanti, A.A. Jr Mironov, J.A. Martinez-Menarguez, O. Martella, A. Fusella, M. Baldassarre, R. Buccione, H.J. Geuze, A.A. Mironov & A. Luini. *Procollagen traverses the Golgi stack without leaving the lumen of cisternae : evidence for cisternal maturation*. Cell, no. 95, pages 993–1003, 1998.
- [Bottoni 02] P. Bottoni & M. Minas. *Workshop on Graph Transformation and Visual Modeling Techniques*. In ICGT '02 : Proceedings of the First International Conference on Graph Transformation, pages 445–449, London, UK, 2002. Springer-Verlag.

- [Bryant 85] R. Bryant & D. Singerman. *Foundations of the theory of maps on surfaces with boundaries*. Quart. J. Math. Oxford Ser. (2), vol. 36, no. 141, pages 17–41, 1985.
- [Calzone 06] L. Calzone, F. Fages & S. Soliman. *BIOCHAM : an environment for modeling biological systems and formalizing experimental knowledge*. Bioinformatics, vol. 22, no. 14, pages 1805–1807, 2006.
- [Cardelli 05] L. Cardelli. *Brane Calculi. Interactions of biological membranes*. In Proc. Computational Methods in Systems Biology, volume 3082 of *Lecture Notes in Bioinformatics*, pages 257–280. Springer, 2005.
- [Cas] CASL. Langage libre de spécification formelle développé par CoFI (Common Framework Initiative). www.caslsoft.com.
- [Charneau 07] S. Charneau. *Étude et application des algèbres géométriques pour le calcul de la visibilité globale dans un espace projectif de dimension $n \geq 2$* . PhD thesis, Université de Poitiers – Laboratoire SIC, Décembre 2007.
- [Chen 07] Rui Chen. *Transformations et Grammaires de Graphes Basées sur l'Opération de Pullback*. PhD thesis, Université Bordeaux 1, Décembre 2007.
- [Cleaveland 77] C.J. Cleaveland & R.C. Uzgalis. *Grammars for programming languages*. Elsevier, New York, 1977.
- [Cohen 93] M.F. Cohen, J. Wallace & P. Hanrahan. *Radiosity and realistic image synthesis*. Academic Press Professional, Inc., San Diego, CA, USA, 1993.
- [Coq] Coq. Langage libre de preuves formelles développé dans le cadre du projet Coq de l'INRIA. pauillac.inria.fr/coq/coq-fra.html.
- [Dittrich 01] P. Dittrich, J. Ziegler & W. Banzhaf. *Artificial chemistries : a review*. Artificial Life, vol. 7, pages 225–275, 2001.
- [Drewes 96] F. Drewes, A. Habel & H.-J. Kreowski. *Hyperedge Replacement Graph Grammars*. In G. Rozenberg, editeur, *Handbook of Graph Transformations*. Vol. I : Foundations. World Scientific, 1996. To appear.
- [Ehrig 99a] H. Ehrig, G. Engels, H.-J. Kreowski & G. Rozenberg, editeurs. *Handbook of graph grammars and computing by graph transformation*, volume 2 : Applications, Languages and Tools. World Scientific, Singapore, 1999.
- [Ehrig 99b] H. Ehrig, H.-J. Kreowski, U. Montanari & G. Rozenberg, editeurs. *Handbook of graph grammars and computing by graph transformation*, volume 3 : Concurrency, Parallelism and Distribution. World Scientific, Singapore, 1999.
- [Ehrig 06] H. Ehrig, K. Ehrig, U. Prange & G. Taentzer. *Fundamentals of algebraic graph transformation*. Monographs in Theoretical Computer Science. Springer, 2006.
- [Elter 94a] H. Elter. *Étude de structures combinatoires pour la représentation de complexes cellulaires*. PhD thesis, Université Louis-Pasteur de Strasbourg, 1994.
- [Elter 94b] H. Elter & L. Fuchs. *Topologie de l'opération d'arrondi d'arêtes*. Revue internationale de CFAO et d'infographie, vol. 9, no. 6, pages 807–829, 1994.
- [Engelfriet 96] J. Engelfriet & G. Rozenberg. *Node Replacement Graph Grammars*. In G. Rozenberg, editeur, *Handbook of Graph Rewriting*, Vol. I : Foundations. World Scientific, 1996.
- [Farquhar 81] M. G. Farquhar & G. E. Palade. *The Golgi apparatus (complex) - 1954-1981 - from artefact to center stage*. J. Cell Biol., no. 91, pages 77–103, 1981.
- [Farquhar 85] M. G. Farquhar. *Progress in unraveling pathways of Golgi traffic*. Ann. Rev. Cell Biol., no. 1, pages 447–488, 1985.
- [Fernando 04] Randima Fernando. *Gpu gems : Programming techniques, tips and tricks for real-time graphics*. Pearson Higher Education, 2004.

- [Gardan 02] Y. Gardan. *CAO : modélisation géométrique*. Techniques de l'ingénieur, vol. HB4, no. H3750, pages 1–24, 2002.
- [Golgi 98] C. Golgi. *Sur la structure des cellules nerveuses des ganglions spinaux*. Arch. Ital. Biol., no. 30, pages 278–286, 1898.
- [Habel 02] A. Habel & D. Plump. *Relabelling in Graph Transformation*. In ICGT '02 : Proceedings of the First International Conference on Graph Transformation, pages 135–147, London, UK, 2002. Springer-Verlag.
- [Hawes 99] C.R. Hawes, F. Brandizzi & A. V. Andreeva. *Endomembranes and vesicles trafficking*. Curr. Opin. in Plant Biol., no. 2, pages 454–461, 1999.
- [Hoffmann 05] B. Hoffmann. *Graph Transformation with Variables*. In H.-J. Kreowski, U. Montanari, F. Orejas, G. Rozenberg & G. Taentzer, éditeurs, Formal Methods in Software and System Modeling (Festschrift for Hartmut Ehrig on the Occasion of his 60th Birthday), volume 3393 of *Lecture Notes in Computer Science*, pages 101–115. Springer-Verlag Heidelberg, 2005.
- [Horna 09] S. Horna, D. Meneveaux, G. Damiand & Y. Bertrand. *Consistency constraints and 3D building reconstruction*. Computer Aided Design, vol. 41, no. 1, pages 13–27, January 2009.
- [Janssens 83] D. Janssens. *Node Label Controlled Graph Grammars*. PhD thesis, Antwerp, 1983.
- [Jensen 01] H.W. Jensen. Realistic image synthesis using photon mapping. A. K. Peters, Ltd., Natick, MA, USA, 2001.
- [Képès 03] F. Képès. *Membrane Traffic in Virtual Reality*. Traffic, vol. 4, no. 1, pages 13–17, January 2003.
- [Képès 04] F. Képès, A. Rambourg & B. Satiat-Jeunemaître. *Morphodynamics of the Secretory Pathway*. International review of cytology, vol. 242, pages 55–120, 2004.
- [Képès 06] F. Képès. Morphogenèse, l'origine des formes, chapitre Morphodynamique de la voie sécrétoire, pages 129–152. Belin, Paris, 2006.
- [Laidlaw 86] D.H. Laidlaw, W.B. Trumbore & J.F. Hughes. *Constructive solid geometry for polyhedral objects*. SIGGRAPH Comput. Graph., vol. 20, no. 4, pages 161–170, 1986.
- [Ledoux 01] F. Ledoux, A. Arnould, P. Le Gall & Y. Bertrand. *Geometric modeling with CASL*. Avril 2001. Recent Trends in Algebraic Development Techniques, 15th International Workshop, WADT 2001, Joint with the CoFI WG Meeting, Genova, Italy, Selected Paper.
- [Ledoux 02a] F. Ledoux. *Étude et spécifications formelles de l'arrondi d'objets géométriques*. PhD thesis, Université d'Évry Val d'Essonne, 2002.
- [Ledoux 02b] F. Ledoux, J.-M. Mota, A. Arnould, C. Dubois, P. Le Gall & Y. Bertrand. *Spécifications formelles du chanfreinage*. Technique et Science Informatiques, vol. 21, no. 8, pages 1–26, 2002.
- [Ledoux 02c] F. Ledoux, J.-M. Mota, A. Arnould, C. Dubois, P. Le Gall & Y. Bertrand. *Spécifications formelles du chanfreinage*. 2002. Technique et Science Informatiques, Volume 21, n°8, p 1-26.
- [Léon 08] P.-F. Léon, X. Skapin & P. Meseure. *Modèle générateur d'évolutions géologiques par animation basée sur la topologie*. Revue Électronique Francophone d'Informatique Graphique (REFIG), Novembre 2008. (en soumission).
- [Lienhardt 89] P. Lienhardt. *Subdivision of n-dimensional spaces and n-dimensional generalized maps*. In SCG'89, pages 228–236. ACM Press, 1989.
- [Lienhardt 94] P. Lienhardt. *n-dimensional generalised combinatorial maps and cellular quasimanifolds*. IJCGA, 1994.

- [Lienhardt 04] P. Lienhardt, X. Skapin & A. Bergey. *Cartesian product of simplicial and cellular structures*. International Journal on Computational Geometry and Applications, vol. 14, no. 3, pages 115–159, Juin 2004.
- [Lin 04] H. X. Lin. *Graph transformation and designing parallel sparse matrix algorithms beyond data dependence analysis*. Sci. Program., vol. 12, no. 2, pages 91–100, 2004.
- [Magaud 08] N. Magaud, J. Narboux & P. Schreck. *Formalizing Projective Plane Geometry in Coq*. In Thomas Sturm, éditeur, Automated Deduction in Geometry (ADG) 2008, pages 1–20, Shanghai Chine, 2008. Thomas Sturm.
- [May] *Maya*. Logiciel commercial de modélisation, d’animation et de rendu 3D développé par Autodesk. Maya est réputé pour son utilisation dans l’industrie cinématographique. www.autodesk.com.
- [Milner 92] Robin Milner. *The Polyadic Pi-calculus (Abstract)*. In CONCUR ’92 : Proceedings of the Third International Conference on Concurrency Theory, page 1, London, UK, 1992. Springer-Verlag.
- [Mironov 97] A. A. Mironov, P. Weidman & A. Luini. *Variations on the intracellular transport theme : maturing cisternae and trafficking tubules*. J. Cell Biol., no. 138, pages 481–484, 1997.
- [Mironov 98] A. Jr. Mironov, A. Luini & A. Mironov. *A synthetic model of intra-Golgi traffic*. FASEB J., no. 12, pages 249–252, 1998.
- [Mollenhauer 91] H. H. Mollenhauer & D. J. Morré. *Perspectives on Golgi apparatus and function*. J. Electr. Microsc. Technique, no. 17, pages 2–14, 1991.
- [Montanari 70] U. Montanari. *Separable Graphs, Planar Graphs and Web Grammars*. Information and Control, vol. 16, pages 243–267, 1970.
- [Murray 08a] J. D. Murray. *Mathematical biology i : An introduction*. Springer, 3 édition, 2008.
- [Murray 08b] J. D. Murray. *Mathematical biology ii : Spatial models and biomedical applications*. Springer, 3 édition, 2008.
- [Oca] *OCaml*. Langage de programmation fonctionnelle développé à l’INRIA. caml.inria.fr.
- [OGL] *OpenGL*. Une bibliothèque pour le développement d’application graphiques. www.opengl.org.
- [Pfaltz 69] John L. Pfaltz & Azriel Rosenfeld. *Web Grammars*. In Int. Joint Confenence on Artificial Intelligence, pages 609–619, 1969.
- [Pharr 05] M. Pharr & R. Fernando. *Gpu gems 2 : Programming techniques for high-performance graphics and general-purpose computation (gpu gems)*. Addison-Wesley Professional, 2005.
- [Pix] *Pixar Animation Studios*. Société de production de films en images de synthèse dont Toy Story (1995) et Wall-E (2008). www.pixar.com.
- [Poudret 05] M. Poudret, A. Arnould & Y. Bertrand. *Un Générateur de Noyaux de Modeleurs*. In 18èmes Journées de l’Association Française d’Informatique Graphique et de l’Association Chapitre Français d’Eurographics (AFIG’05), November 2005.
- [Poudret 07a] M. Poudret, J.-P. Comet, P. Le Gall, A. Arnould & P. Meseure. *Exploring Topological Modelling to Discriminate Models of Golgi Apparatus Dynamics*. In European Conference on Complex Systems (ECCS’07), October 2007.
- [Poudret 07b] M. Poudret, J.-P. Comet, P. Le Gall, A. Arnould & P. Meseure. *Topology-based Geometric Modelling for Biological Cellular Processes*. In 1st International conference on Langage and Automata Theory and Applications (LATA’07), Tarragona, Spain, March 2007.

- [Poudret 08a] M. Poudret, A. Arnould, J.-P. Comet & P. Le Gall. *Graph Transformation for Topology Modelling*. In 4th International Conference on Graph Transformation (ICGT'08), numéro 5214 in LNCS, pages 147–161, September 2008.
- [Poudret 08b] M. Poudret, J.-P. Comet, P. Le Gall, F. Képès, A. Arnould & P. Meseure. *Topology-based Abstraction of Complex Biological Systems : Application to the Golgi Apparatus*. Theory in Biosciences, vol. 127, no. 2, pages 79–88, January 2008.
- [Poudret 08c] M. Poudret, J.-P. Comet, P. Le Gall, F. Képès, A. Arnould, P. Meseure, J.-M. Verbatz & A. Rambourg. Proceeding of the lille spring school on modelling complex biological systems in the context of genomics., chapitre Toward a computer-aided method, pages 89–104. EDP SCIENCES, January 2008.
- [Pratt 71] Terrence W. Pratt. *Pair Grammars, Graph Languages and String-to-Graph Translations*. Journal of Computer and System Sciences, vol. 5, pages 560–595, 1971.
- [Rambourg 69] A. Rambourg. *L'appareil de Golgi : examen en microscopie électronique de coupes épaisses (0,5 – 1 μ), colorées par le mélange chlorhydrique-phosphotungstique*. C. R. Acad. Sci. Paris, no. 269, pages 2125–2127, 1969.
- [Rambourg 79] A. Rambourg, Y. Clermont & L. Hermo. *Three-dimensional architecture of the Golgi apparatus in the Sertoli cell of the rat*. Am. J. Anat., no. 154, pages 455–476, 1979.
- [Regev 04] A. Regev, E. M. Panina, W. Silverman, L. Cardelli & E. Shapiro. *BioAmbients : an abstraction for biological compartments*. Theor. Comput. Sci., vol. 325, no. 1, pages 141–167, 2004.
- [Requicha 82] A.A.G. Requicha & H.B. Voelcker. *Solid Modeling : A Historical Summary and Contemporary Assessment*. IEEE Comput. Graph. Appl., vol. 2, no. 2, pages 9–24, 1982.
- [Robinson 98] D.G. Robinson, G. Hinz & S. E. H. Holstein. *The molecular characterization of transport vesicles*. Plant Molecular Biology, no. 38, pages 49–76, 1998.
- [Rothman 94] J. E. Rothman. *Mechanisms of intracellular protein transport*. Nature, no. 372, pages 55–63, 1994.
- [Rozenberg 97] G. Rozenberg, editeur. *Handbook of graph grammars and computing by graph transformation, volume 1 : Foundations*. World Scientific, Singapore, 1997.
- [Saraste 07] J. Saraste & B. Goud. *Functional Symmetry of Endomembranes*. Mol. Biol. Cell, vol. 18, no. 4, pages 1430–1436, 2007.
- [Skapin 01a] X. Skapin. *Utilisation du produit cartésien en modélisation géométrique 4D pour l'animation*. PhD thesis, Université de Poitiers, 2001.
- [Skapin 01b] X. Skapin & P. Lienhardt. *Using Cartesian Product for Animation*. 2001. Journal of Visualization and Computer Animation, Vol. 12, n° 3, pp. 131-144.
- [Spicher 06] A. Spicher. *Transformation de collections topologiques de dimension arbitraire. Application à la modélisation de systèmes dynamiques*. PhD thesis, Université d'Évry-Val-d'Essonne U.F.R. Sciences Fondamentales et Appliquées, Décembre 2006.
- [Sutherland 63] I. Sutherland. *Sketchpad : a Man-Machine Graphical Communication System*. PhD thesis, Massachusetts Institute of Technology, 1963.
- [Syropoulos 01] A. Syropoulos. *Mathematics of Multisets*. In In Multiset Processing, pages 347–358. Springer, 2001.
- [Terraz 09] O. Terraz, G. Guimberteau, S. Mérillou, D. Plemenos & D. Ghazanfarpour. *3Gmap L-systems : an application to the modelling of wood*. Vis. Comput., vol. 25, no. 2, pages 165–180, 2009.
- [Tutte 84] W. Tutte. Graph theory, volume 21 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, 1984.

- [Van Eetvelde 04] N. Van Eetvelde & D. Janssens. *Extending Graph Rewriting for Refactoring*. In ICGT, pages 399–415, 2004.
- [Vidil 03] F. Vidil & G. Damiand. Moka. www.sic.sp2mi.univ-poitiers.fr/moka/, 2003.
- [Wattenbreg 91] B. W. Wattenbreg. *Analysis of protein transport through the Golgi in a reconstituted cell-free system*. J. Electr. Microsc. Technique, no. 17, pages 150–131, 1991.
- [Weiler 88] K. Weiler. *The Radial Edge Structure : A Topological Representation for Non-Manifold Geometric Boundary Modeling*. In Geometric Modeling for CAD Applications : Selected and Expanded Papers from the Ifip Wg 5.2 Working Conference, 1988.
- [Wik] *Wikipédia*. fr.wikipedia.org.
- [Zampelli 07] S. Zampelli, Y. Deville, C. Solnon, S. Sorlin & P. Dupont. *Filtrage pour l'isomorphisme de sous-graphe*. In F. Fages, editeur, Journées Francophones de Programmation par Contraintes (JFPC), June 2007.
- [Zienkiewicz 05] O. C. Zienkiewicz, R. L. Taylor & J. Z. Zhu. *The finite element method : Its basis and fundamentals*. Butterworth-Heinemann, 6 edition, 2005.

Transformations de graphes pour les opérations topologiques en modélisation géométrique - Application à l'étude de la dynamique de l'appareil de Golgi

Résumé Dans cette thèse, qui s'inscrit dans l'étude de la modélisation géométrique via les méthodes formelles, nous proposons un langage graphique à base de règles dédié à la description des opérations topologiques des cartes généralisées. Notre langage est fondé sur la théorie des transformations de graphes. Dans nos règles, les variables permettent d'abstraire les cellules topologiques (sommets, arêtes, faces, volumes, etc.) manipulées dans les opérations topologiques. Nous avons défini des critères syntaxiques sur les règles assurant que les objets obtenus par application des règles satisfont les contraintes de cohérence des cartes généralisées. La conception de ce langage a été motivée par l'étude de la dynamique de l'appareil de Golgi. Il est connu que dans cette organelle, la topologie des compartiments joue un rôle essentiel. Néanmoins, la structure globale de l'appareil de Golgi reste encore méconnue. Plusieurs hypothèses de fonctionnement sont ainsi avancées par les biologistes. Notre langage à base de règles fournit un cadre pour la simulation puis la comparaison de ces différentes hypothèses d'appareil de Golgi.

Mots clés : modélisation géométrique à base topologique, cartes généralisées, transformations de graphes, réécriture topologique, simulation en biologie cellulaire.

Graphs transformations for the topological operations of the geometric modeling - Application to the study of the Golgi apparatus dynamics

Abstract This PhD thesis is in line with the study of geometric modeling by the means of formal methods. We propose a graphical rule-based language dedicated to the description of the topological operations of the generalized maps. Our language is based on the graphs transformations theory. In our rules, variables allow one to abstract the topological cells (vertices, edges, faces, volumes, etc.) handled in the topological operations. We have defined some syntactic criteria over rules, which ensure that the objects computed by rules application satisfy the consistency constraints of the generalized maps. The conception of this language has been motivated by the study of the Golgi apparatus dynamics. It is known that in this organelle, the topology of the compartments plays a decisive role. Nevertheless, the global Golgi apparatus structure is still unknown. The biologists thus propose several functional hypotheses. Our rule-based language provides a framework to simulate and the to compare these different hypotheses.

Keywords : topology-based geometric modeling, generalized maps, graphs transformations, topological rewriting, simulations of cell biology.

Discipline : informatique.

Mathieu Poudret - Laboratoire XLIM-SIC, UMR CNRS 6172, Université de Poitiers
Bât. SP2MI - Téléport 2, Bd Marie et Pierre Curie - BP 30179 - 86962 Futuroscope Cedex

Programme d'Épigénomique, Genopole, CNRS UPS3201, PRES UniverSud Paris, UEVE
Genopole Campus 1 - Genavenir 6 5 rue Henri Desbruères - 91030 EVRY cedex